Learning Hierarchical Generative Models

Russ Salakhutdinov

Department of Statistics and Computer Science
University of Toronto

Machine Learning's Successes

- Computer Vision:
  - Image inpainting/denoising, segmentation
  - object recognition/detection, scene understanding
- Information Retrieval / NLP:
  - Text, audio, and image retrieval
  - Parsing, machine translation, text analysis
- Speech processing:
  - Speech recognition, voice identification
- Robotics:
  - Autonomous car driving, planning, control
- Computational Biology
- Cognitive Science.

Mining for Structure

Massive increase in both computational power and the amount of data available from web, video cameras, laboratory measurements.

Images & Video

Text & Language

Speech & Audio

Gene Expression

Deep Generative Models that support inferences and discover structure at multiple levels.

Mostly Unlabeled

- Develop statistical models that can discover underlying structure, cause, or statistical correlation from data in **unsupervised** or **semi-supervised** way.
- Multiple application domains.

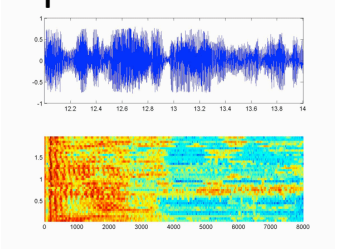
Mining for Structure

Massive increase in both computational power and the amount of data available from web, video cameras, laboratory measurements.

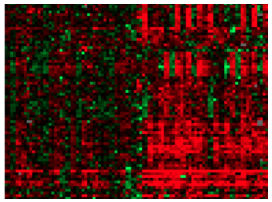
Images & Video

Text & Language

Speech & Audio

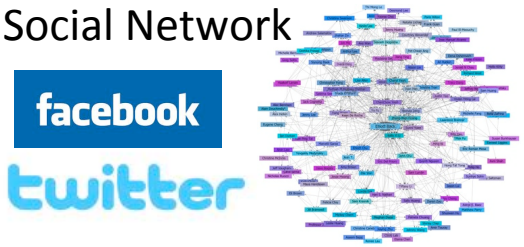


Gene Expression

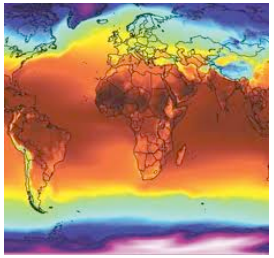


Product Recommendation

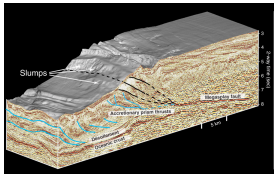
Relational Data/ Social Network



Climate Change



Geological Data



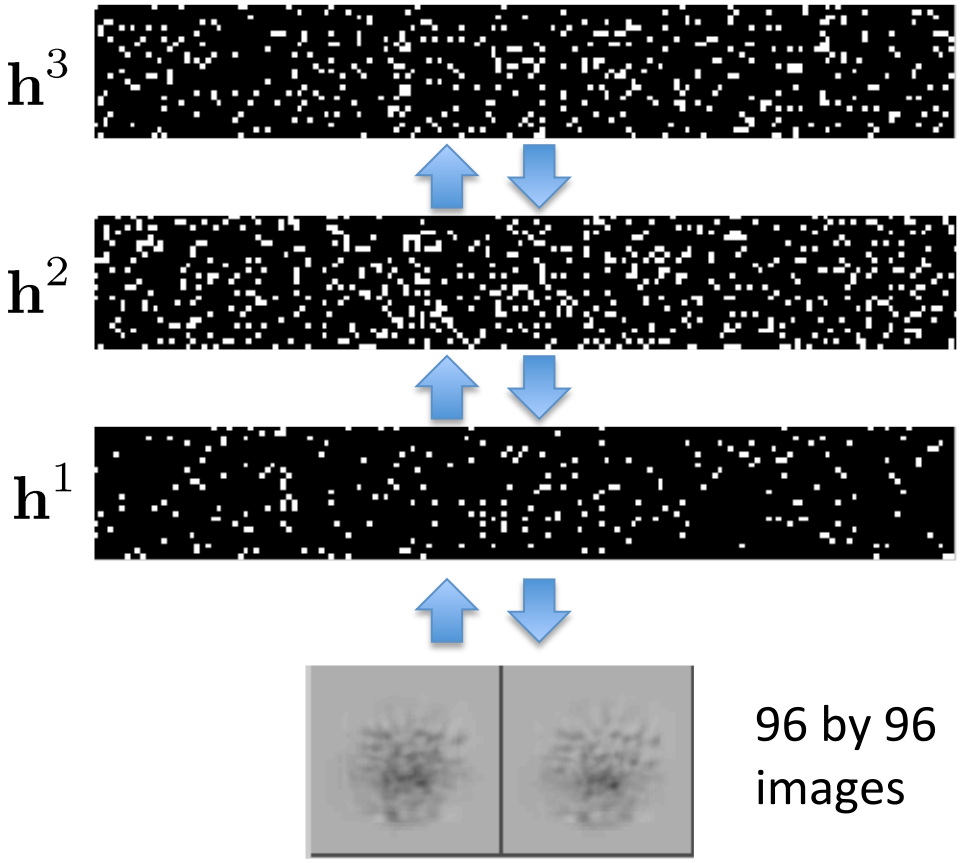
Mostly Unlabeled

- Develop statistical models that can discover underlying structure, cause, or statistical correlation from data in **unsupervised** or **semi-supervised** way.
- Multiple application domains.

Deep Generative Model

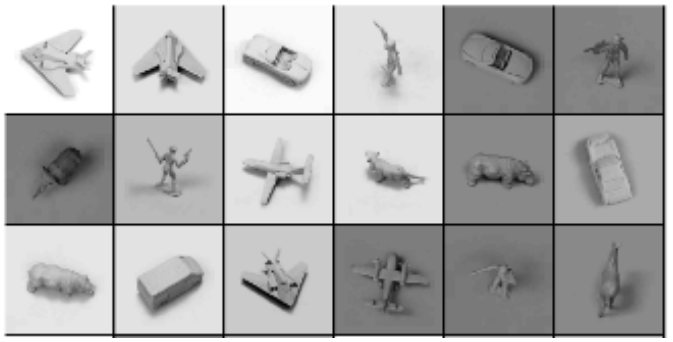
Deep Boltzmann Machine

Gaussian-Bernoulli Markov Random Field



12,000 Latent Variables

Model P(image)



24,000 Training Images

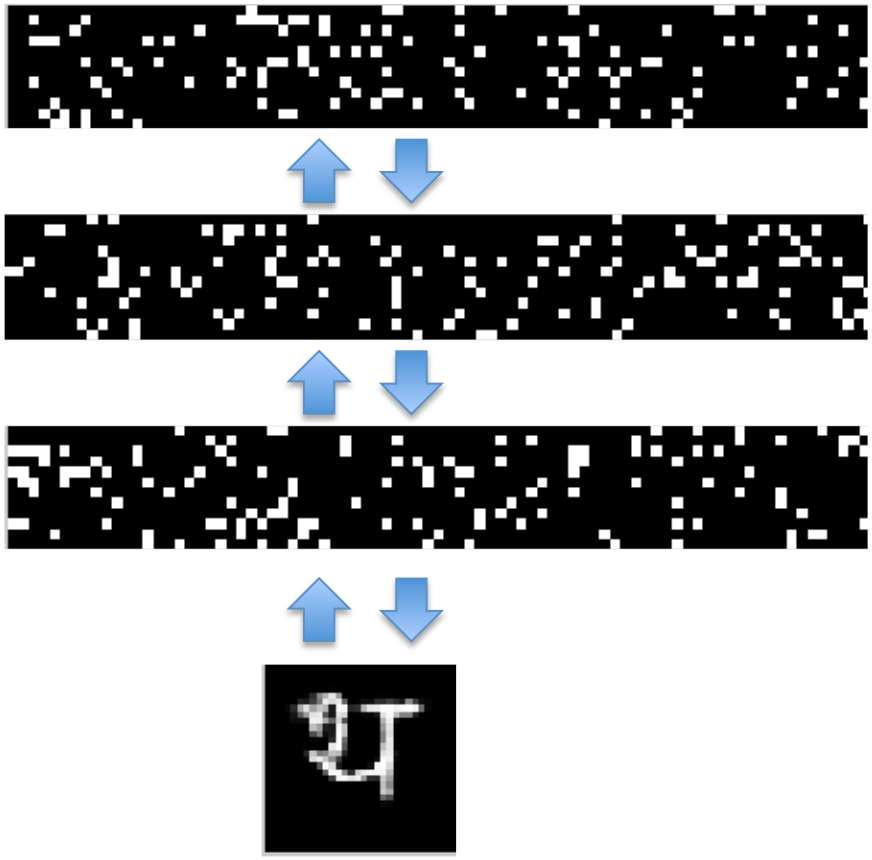
Stereo pair

96 by 96 images

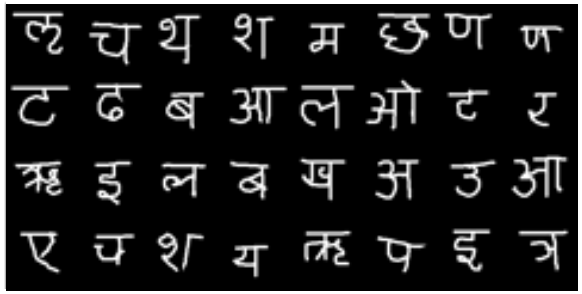
(Salakhutdinov, 2008; Salakhutdinov & Hinton, AI & Statistics 2009)

Deep Generative Model

Sanskrit



Model P(image)

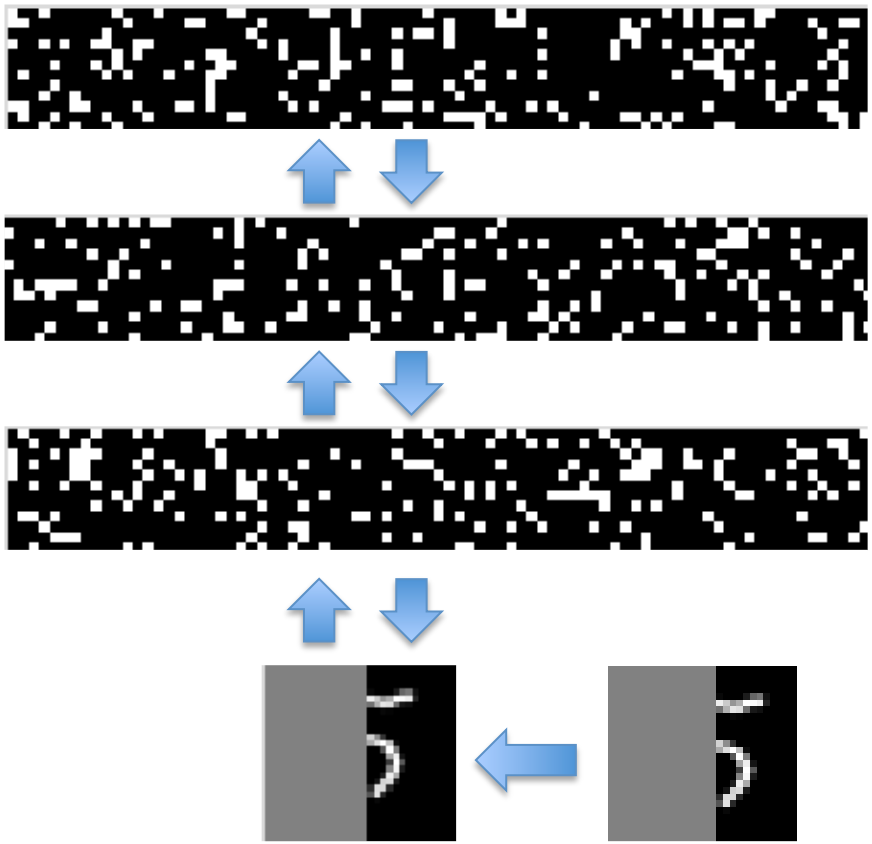


25,000 characters from 50 alphabets around the world.

- 3,000 hidden variables
- 784 observed variables (28 by 28 images)
- Over 2 million parameters

Bernoulli Markov Random Field

Deep Generative Model

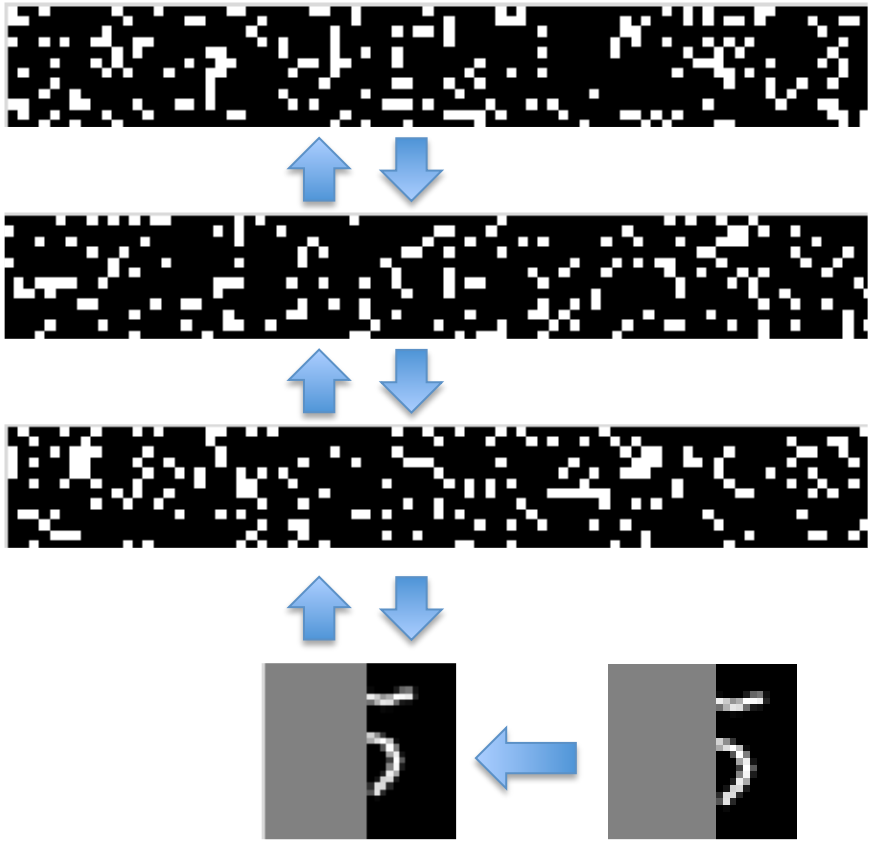


Conditional Simulation

P(image|partial image)

Bernoulli Markov Random Field

Deep Generative Model



P(image|partial image)

Conditional Simulation

Why so difficult?

28

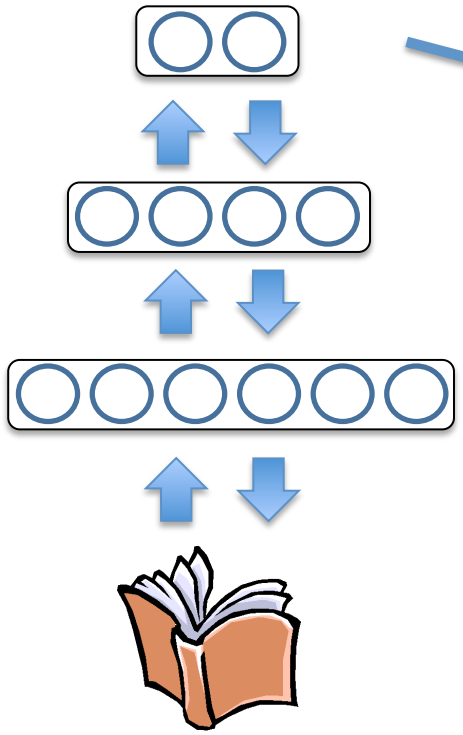
28

$2^{28\times 28}$ possible images!

Bernoulli Markov Random Field

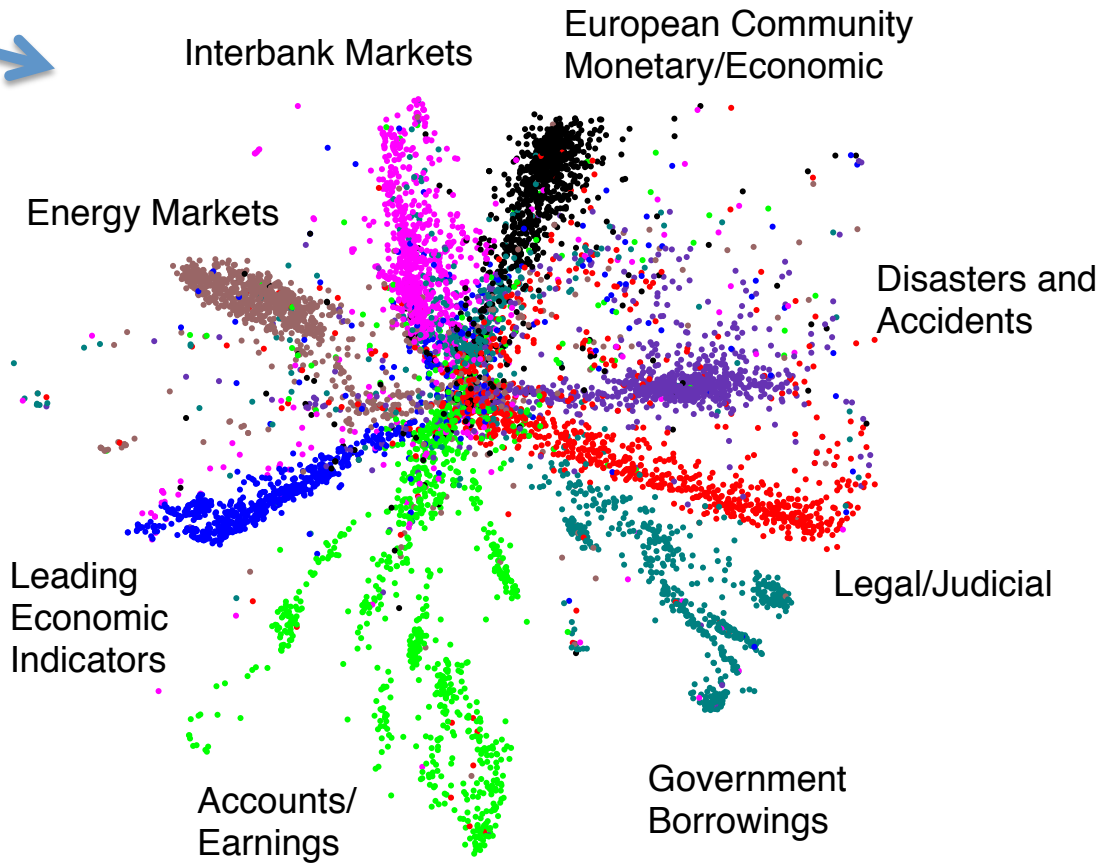
Deep Generative Model

Model P(document)



Bag of words

Reuters dataset: 804,414
newswire stories: **unsupervised**



(Hinton & Salakhutdinov, Science 2006)

Talk Roadmap

Part 1: Deep Networks

- Introduction, Graphical Models.
- Restricted Boltzmann Machines: Learning low-level features.
- Deep Belief Networks: Learning Part-based Hierarchies.

Part 2: Deep Boltzmann Machines.

- Inference and Learning
- Advanced Deep Models

Inference Problem

- Given a dataset $\mathcal{D} = \{x_1, x_2, ..., x_n\}$

- Bayes Rule:

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})}$$

$P(\mathcal{D}\|\theta)$	Likelihood function
$P(\theta)$	Prior probability of parameters
$P(\theta\|\mathcal{D})$	Posterior distribution over parameters

- Computing posterior distribution is known as **inference** problem.

However,

$$P(\mathcal{D}) = \int P(\mathcal{D}|\theta)P(\theta)d\theta$$

- This integral can be very high-dimensional and difficult to compute.

Prediction

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})}$$

$P(\mathcal{D}\|\theta)$	Likelihood function
$P(\theta)$	Prior probability of parameters
$P(\theta\|\mathcal{D})$	Posterior distribution over parameters

- **Prediction**: Given data, computing conditional probability of a new data point x* requires computing the following integral:

$$\begin{aligned} P(x^*|\mathcal{D}) &= \int P(x^*|\theta)P(\theta|D)d\theta \\ &= \mathbb{E}_{P(\theta|D)}[P(x^*|\theta)] \end{aligned}$$

  which is sometimes called **predictive distribution**.

- Computing predictive distribution requires posterior.

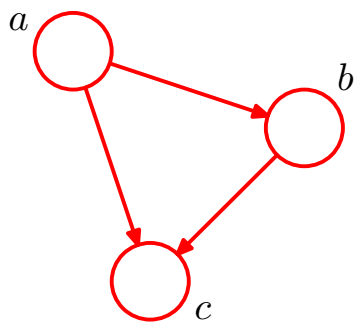
Computational Challenges

- Computing marginal likelihoods often requires computing very high-dimensional integrals.
- Computing posterior distributions (and hence predictive distributions) is often analytically intractable.

- Next: Graphical Models.

Graphical Models

Graphical Models: Powerful framework for representing dependency structure between random variables.



- The joint probability distribution over a set of random variables.
- The graph contains a set of nodes (vertices) that represent random variables, and a set of links (edges) that represent dependencies between those random variables.

- The joint distribution over all random variables decomposes into a **product of factors**, where each factor depends on a subset of the variables.

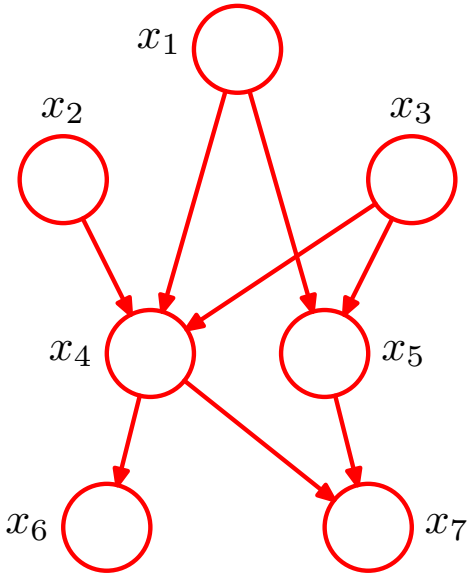
Two type of graphical models:

- **Directed** (Bayesian networks)
- **Undirected** (Markov random fields, Boltzmann machines)

Hybrid graphical models that combine directed and undirected models, such as Deep Belief Networks, Hierarchical-Deep Models.

Directed Graphical Models

Directed graphs are useful for expressing causal relationships between random variables.



• The joint distribution defined by the graph is given by the **product of a conditional distribution for each node conditioned on its parents.**

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$

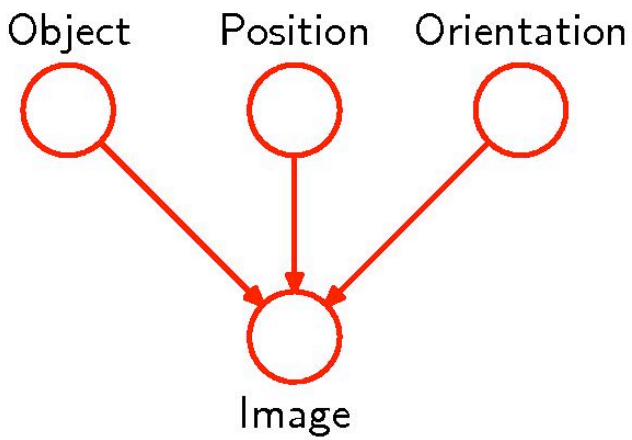
• For example, the joint distribution over x1,..,x7 factorizes:

$$p(\mathbf{x}) = p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3)p(x_5|x_1,x_3)p(x_6|x_4)p(x_7|x_4,x_5)$$

Directed acyclic graphs, or *DAGs*.

Directed Graphical Models

Example: Generative model of an image:



• Object identity (discrete variable) and the position and orientation (continuous variables) have **independent prior probabilities**.

• The image has a probability distribution that depends on the object identity, position, and orientation **(likelihood function)**.

The joint distribution:

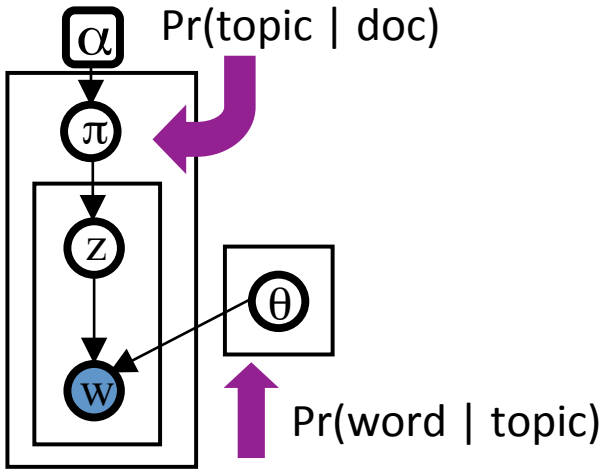
$$P(Im, Ob, Po, Or) = \underbrace{P(Im|Ob, Po, Or)}_{\text{Likelihood}} \underbrace{P(Ob)P(Po)P(Or)}_{\text{Prior}}$$

Inference: Computing posterior:

$$P(Ob, Po, Or|Im) = \frac{1}{\underbrace{P(Im)}_{\text{Marginal likelihood: Often difficult to compute}}} P(Im|Ob, Po, Or)P(Ob)P(Po)P(Or)$$

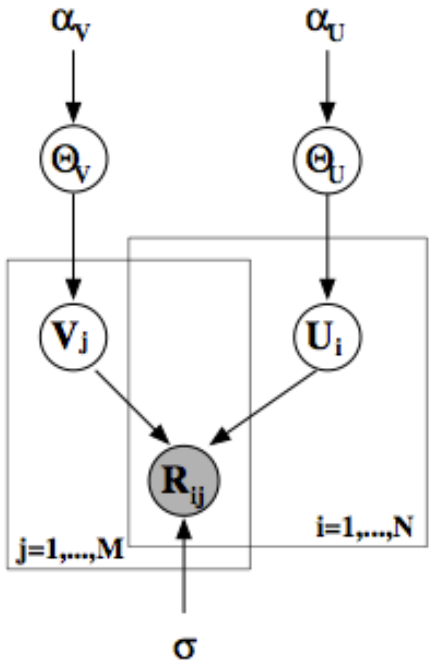
Popular Models

Latent Dirichlet Allocation



• One of the popular models for modeling word count vectors. We will see this model later.

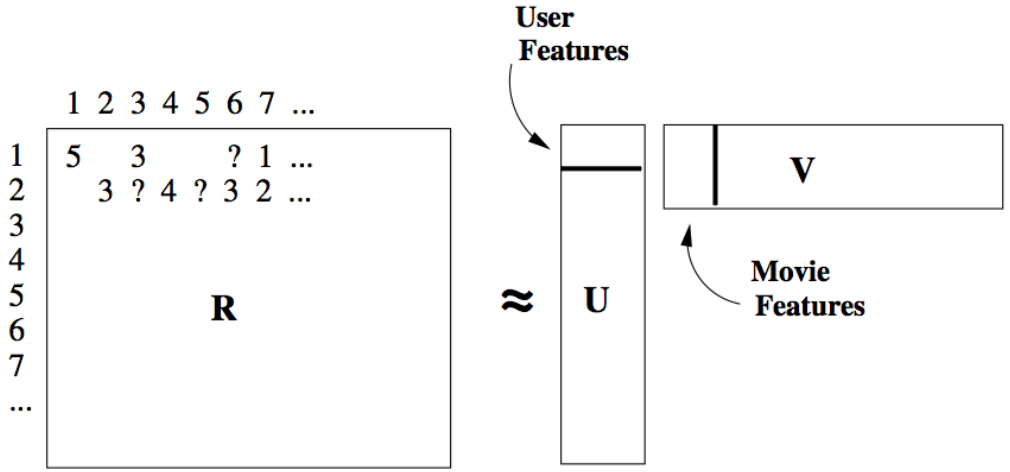
Probabilistic Matrix Factorization



• One of the popular models for collaborative filtering applications. Part of the winning solution in the Netflix contest.

Bayesian Matrix Factorization

- Let us first look at some examples.



- We have N users, M movies, and integer rating values from 1 to K.
- Let r_{ij} be the rating of user i for movie j, and $U \in R^{D \times \mathcal{N}}$, and $V \in R^{D \times \mathcal{M}}$ be latent user and movie feature matrices:

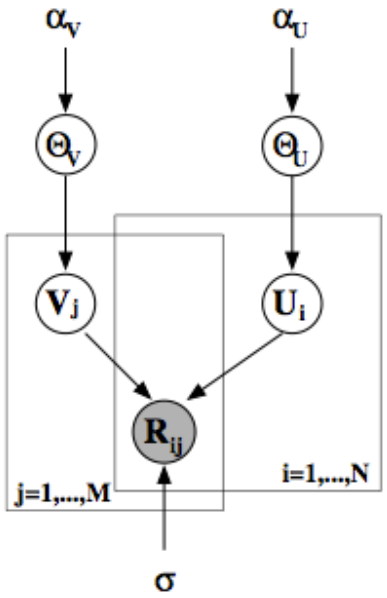
$$R \approx U^T V.$$

- Our goal is to predict missing values (missing ratings).

(Salakhutdinov & Mnih, ICML 2008)

Bayesian Matrix Factorization

• We can define a probabilistic bilinear model with Gaussian observation noise:



$$p(r_{ij}|U, V, \sigma^2) = \mathcal{N}(r_{ij}|u_i^T v_j, \sigma^2).$$

• We can place Gaussian priors over latent variables:

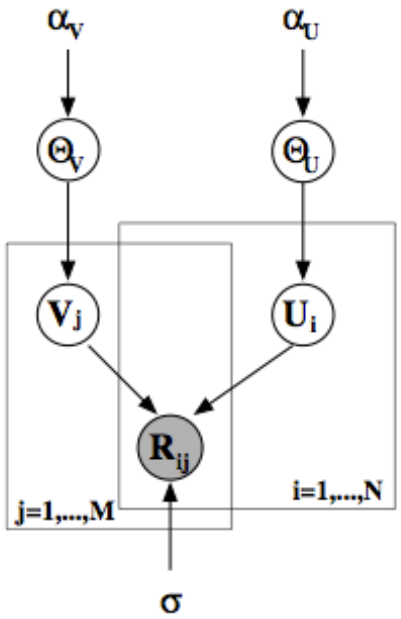
$$p(U|\mu_U, \Lambda_U) = \prod_{i=1}^{N} \mathcal{N}(u_i|\mu_U, \Lambda_U^{-1}),$$

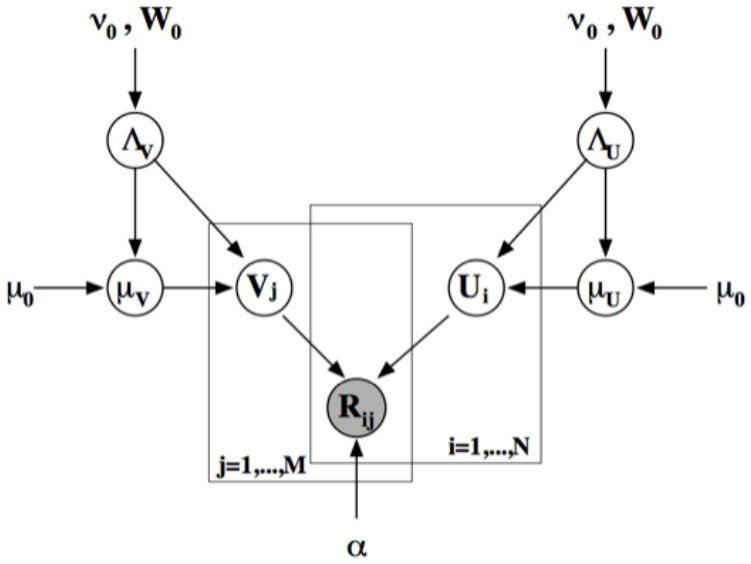
$$p(V|\mu_V, \Lambda_V) = \prod_{j=1}^{M} \mathcal{N}(v_j|\mu_V, \Lambda_V^{-1}).$$

• **Hierarchical Prior**: introduce Gaussian-Wishart priors over the user and movie hyper-parameters:

$$\Theta_U = \{\mu_U, \Lambda_U\}, \ \Theta_V = \{\mu_V, \Lambda_V\}.$$

Bayesian Matrix Factorization





$$p(r_{ij}|U,V,\sigma^2) = \mathcal{N}(r_{ij}|u_i^T v_j, \sigma^2).$$

$$\Theta_U = \{\mu_U, \Lambda_U\},$$

$$\Theta_V = \{\mu_V, \Lambda_V\}.$$

$$p(U|\mu_U,\Lambda_U) = \prod_{i=1}^{N} \mathcal{N}(u_i|\mu_U,\Lambda_U^{-1}),$$

$$p(V|\mu_V,\Lambda_V) = \prod_{j=1}^{M} \mathcal{N}(v_j|\mu_V,\Lambda_V^{-1}).$$

Predictive Distribution

• Consider predicting a rating r_{ij}^* for user i and query movie j:

$$p(r_{ij}^*|R) = \iint p(r_{ij}^*|u_i, v_j)\underbrace{p(U, V, \Theta_U, \Theta_V|R)}_{\text{Posterior over parameters and hyperparameters}}d\{U, V\}d\{\Theta_U, \Theta_V\}$$

• Exact evaluation of this predictive distribution is analytically intractable.

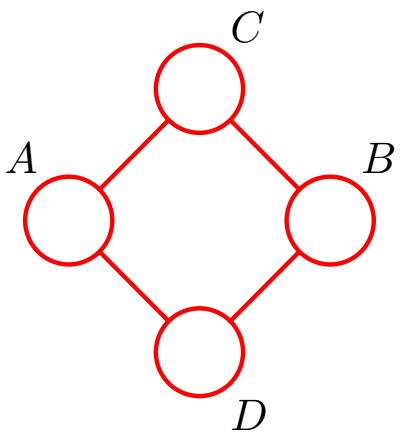
• Posterior distribution over parameters and hyper-parameters is complicated and does not have a closed-form expression.

• Need to approximate.

• One option would be to approximate the posterior using factorized distribution and use variational framework.

• Alternative would be to resort to Monte Carlo methods.

Markov Random Fields



$$p(\mathbf{x}) = \frac{1}{\mathcal{Z}} \prod_C \phi_C(x_C)$$

- Each potential function is a mapping from joint configurations of random variables in a clique to non-negative real numbers.
- The choice of potential functions is not restricted to having specific probabilistic interpretations.

Potential functions are often represented as exponentials:

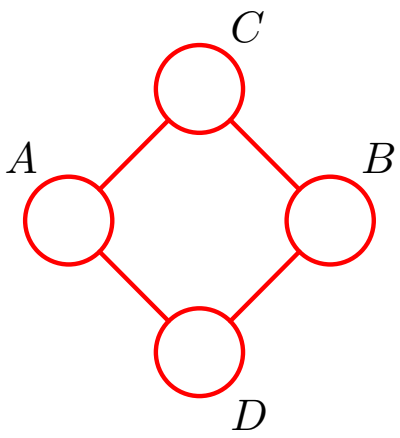
$$p(\mathbf{x}) = \frac{1}{\mathcal{Z}} \prod_C \phi_C(x_C) = \frac{1}{\mathcal{Z}} \exp(-\sum_C E(x_c)) = \underbrace{\frac{1}{\mathcal{Z}} \exp(-E(\mathbf{x}))}_{\text{Boltzmann distribution}}$$

where E(x) is called an energy function.

- Suppose x is a binary random vector with $x_i \in \{+1, -1\}$.
- If x is 100-dimensional, we need to sum over 2^{100} terms!

Computing Z is often very hard. This represents a major limitation of undirected models.

Markov Random Fields



$$p(\mathbf{x}) = \frac{1}{\mathcal{Z}} \prod_C \phi_C(x_C)$$

- Each potential function is a mapping from joint configurations of random variables in a clique to non-negative real numbers.
- The choice of potential functions is not restricted to having specific probabilistic interpretations.

Potential functions are often represented as exponentials:

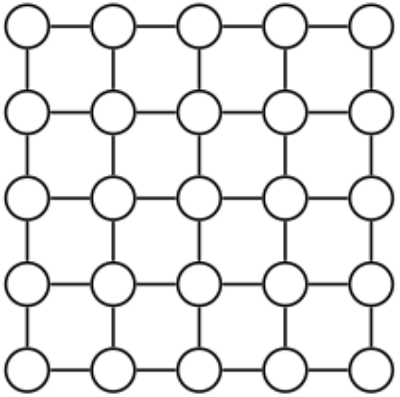
$$p(\mathbf{x}) = \frac{1}{\mathcal{Z}} \prod_C \phi_C(x_C) = \frac{1}{\mathcal{Z}} \exp(-\sum_C E(x_c)) = \underbrace{\frac{1}{\mathcal{Z}} \exp(-E(\mathbf{x}))}_{\text{Boltzmann distribution}}$$

where E(x) is called an energy function.

Compare to computing posterior:

$$P(\theta|\mathcal{D}) = \tfrac{1}{P(\mathcal{D})} P(\mathcal{D}|\theta)P(\theta) \quad \text{where} \quad P(\mathcal{D}) = \int P(\mathcal{D},\theta)d\theta$$

Maximum Likelihood Learning



Consider binary pairwise MRF:

$$P_\theta(\mathbf{x}) = \frac{1}{\mathcal{Z}(\theta)} \exp\Big(\sum_{ij \in E} x_i x_j \theta_{ij} + \sum_{i \in V} x_i \theta_i\Big)$$

Given a set of *i.i.d.* training examples $\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$, we want to learn model parameters θ.

Maximize log-likelihood objective: $L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log P_\theta(\mathbf{x}^{(n)})$

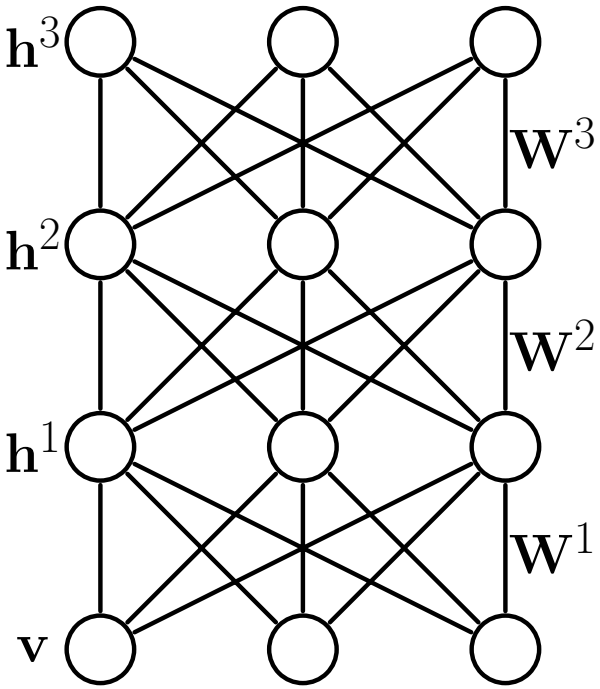
Derivative of the log-likelihood:

$$\frac{\partial L(\theta)}{\partial \theta_{ij}} = \frac{1}{N} \sum_n [x_i^{(n)} x_j^{(n)}] - \underbrace{\sum_{\mathbf{x}} [x_i x_j P_\theta(\mathbf{x})]}_{} = \mathrm{E}_{P_{data}}[x_i x_j] - \mathrm{E}_{P_\theta}[x_i x_j]$$

Difficult to compute: exponentially many configurations

MRFs with Latent Variables

For many interesting real-world problems, we need to introduce hidden or latent variables.



- Our random variables will contain both visible and hidden variables x=(v,h).
- Probability of observed input is given by marginalizing out the states of hidden variables:

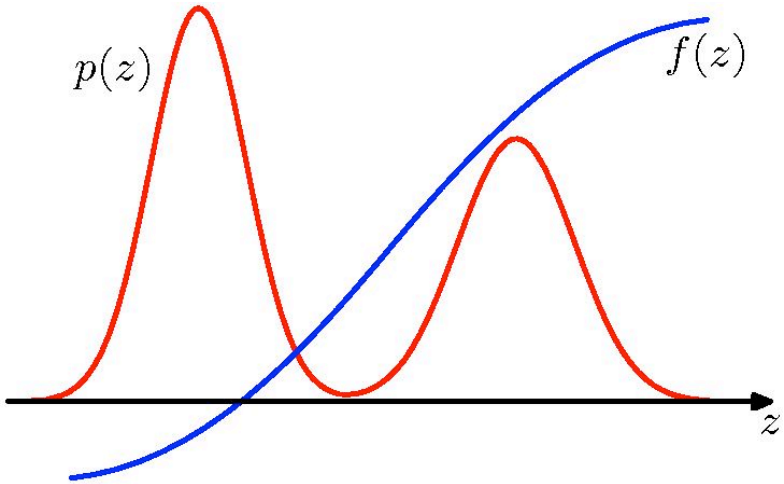
$$p(\mathbf{v}) = \frac{1}{\mathcal{Z}} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))$$

- In general computing both partition function and summation over hiddens will be intractable, except for special cases.
- Parameter learning becomes a very challenging task.

Deep Networks have to deal with this intractability.

Inference Problem

• For most situations, we will be interested in evaluating expectations (for example in order to make predictions):



$$\mathbb{E}[f] = \int f(\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z}.$$

where the integral will be replaced with summation in case of discrete variables.

• We will often use the following notation: $p(\mathbf{z}) = \dfrac{\tilde{p}(\mathbf{z})}{\mathcal{Z}}$.

• We can evaluate $\tilde{p}(\mathbf{z})$ pointwise but cannot evaluate $\mathcal{Z}$.

- Posterior distribution: $p(\theta|\mathcal{D}) = \dfrac{1}{p(\mathcal{D})}p(\mathcal{D}|\theta)p(\theta).$
- Markov Random Fields: $p(\mathbf{x}) = \dfrac{1}{\mathcal{Z}}\exp(-E(\mathbf{x})).$

Variational Inference

- Approximate intractable distribution $P(\theta|\mathcal{D})$ with simpler, tractable distribution $Q(\theta)$.

$$\begin{aligned}
\ln P(\mathcal{D}) &= \ln \int P(\mathcal{D}|\theta)P(\theta)d\theta = \ln \int Q(\theta)\frac{P(\mathcal{D},\theta)}{Q(\theta)}d\theta \\
&\geq \int Q(\theta)\ln\frac{P(\mathcal{D},\theta)}{Q(\theta)}d\theta = \int Q(\theta)\ln p(\mathcal{D},\theta)d\theta + \underbrace{\int Q(\theta)\ln\frac{1}{Q(\theta)}d\theta}_{\text{Entropy Functional}} \\
&= \ln P(\mathcal{D}) - \mathrm{KL}(Q(\theta)||P(\theta|\mathcal{D}))
\end{aligned}$$

Variational Lower Bound

where $\mathrm{KL}(Q||P)$ is a Kullback-Leibler divergence – a non-symmetric measure of the difference between two distributions Q and P:

$$\mathrm{KL}(Q||P) = \int Q(x)\ln\frac{Q(x)}{P(x)}dx$$

Variational Inference

• Approximate intractable distribution $P(\mathcal{D}|\theta)$ with simpler, tractable distribution $Q(\theta)$.

• Variational Lower-bound:

$$\ln P(\mathcal{D}) \geq \ln P(\mathcal{D}) - \mathbf{KL}(Q(\theta)||P(\theta|\mathcal{D}))$$

• The goal of variational inference is to maximize the variational lower-bound with respect to approximate Q distribution, i.e minimize the KL term.

Mean-Field Approximation

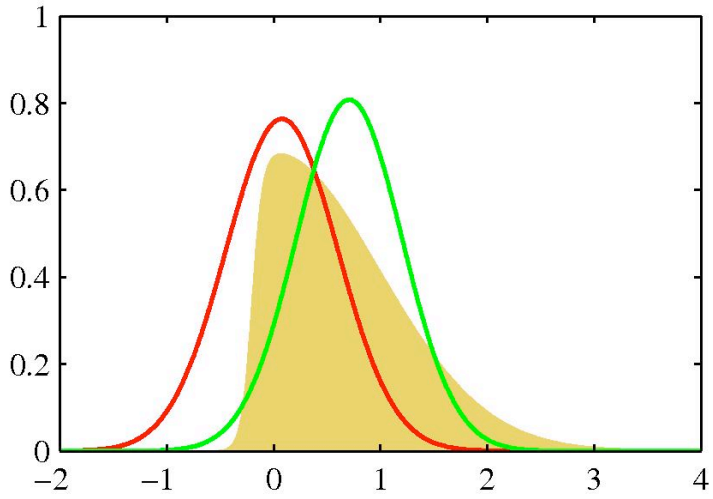
- We can choose a fully factorized distribution: $Q(\theta) = \prod_{i=1}^{D} Q_i(\theta_i)$.
  This is known as a **mean-field approximation**.

- The variational lower-bound takes form:

$$\begin{aligned}
\mathcal{L}(Q) &= \int Q(\theta) \ln \frac{P(\mathcal{D},\theta)}{Q(\theta)} d\theta = \int Q(\theta) \ln p(\mathcal{D},\theta) d\theta + \int Q(\theta) \ln \frac{1}{Q(\theta)} d\theta \\
&= \int Q_j(\theta_j) \underbrace{\left[\ln P(\mathcal{D},\theta) \prod_{i \neq j} Q_i(\theta_i) d\theta_i \right]}_{\mathbb{E}_{i \neq j}[\ln P(\mathcal{D},\theta)]} \theta_j + \sum_i \int Q_i(\theta_i) \ln \frac{1}{Q_i(\theta_i)} d\theta_i
\end{aligned}$$

- Suppose that we keep $\{Q_{i \neq j}\}$ fixed and maximize the bound w.r.t. all possible forms for the distribution $Q_j(\theta_j)$.

Mean-Field Approximation



The original distribution (yellow), along with Laplace (red), and variational (green) approximations.

• By maximizing the bound, we obtain a general form:

$$Q_j^*(\theta_j) = \frac{\exp\left(\mathbb{E}_{i\neq j}[\ln P(\mathcal{D},\theta)]\right)}{\int \exp\left(\mathbb{E}_{i\neq j}[\ln P(\mathcal{D},\theta)]\right) d\theta_j}$$

• Iterative procedure: Initialize all $Q_j(\theta_j)$ and then iterate through the factors replacing each in turn with a revised estimate.

• Convergence is guaranteed (see Bishop, chapter 10).

Talk Roadmap

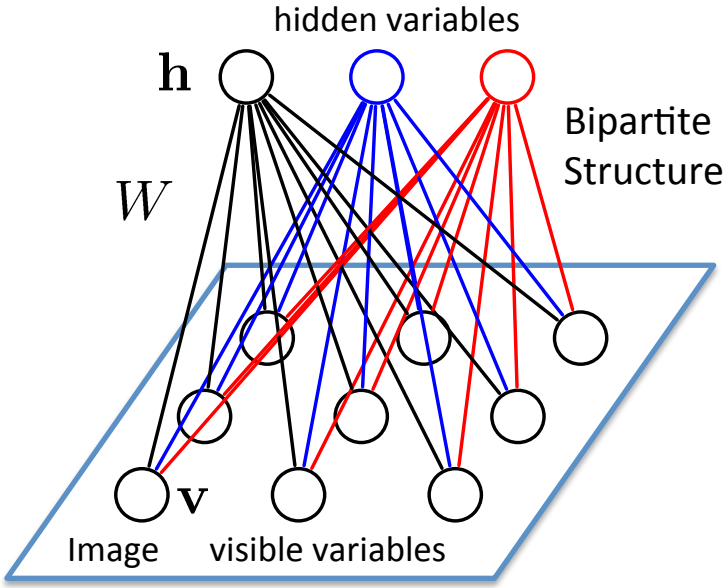
Part 1: Deep Networks

- Introduction, Graphical Models.
- Restricted Boltzmann Machines: Learning low-level features.
- Deep Belief Networks: Learning Part-based Hierarchies.

Part 2: Deep Boltzmann Machines.

- Inference and Learning
- Advanced Deep Models

Restricted Boltzmann Machines



Stochastic binary visible variables $\mathbf{v} \in \{0,1\}^D$ are connected to stochastic binary hidden variables $\mathbf{h} \in \{0,1\}^F$.

The energy of the joint configuration:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{ij} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j$$

$\theta = \{W, a, b\}$ model parameters.

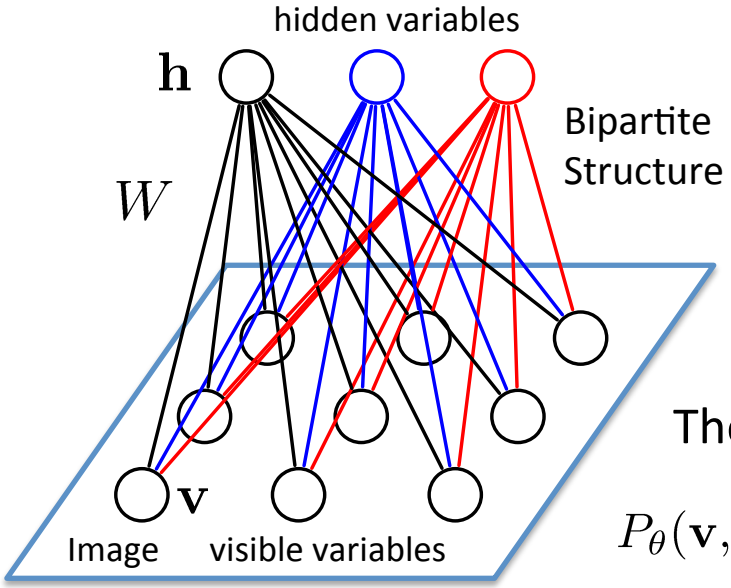
Probability of the joint configuration is given by the Boltzmann distribution:

$$P_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\big(-E(\mathbf{v}, \mathbf{h}; \theta)\big) = \underbrace{\frac{1}{\mathcal{Z}(\theta)}}_{\text{partition function}} \prod_{ij} \underbrace{e^{W_{ij} v_i h_j}}_{\text{potential functions}} \prod_i e^{b_i v_i} \prod_j e^{a_j h_j}$$

$$\mathcal{Z}(\theta) = \sum_{\mathbf{h}, \mathbf{v}} \exp\big(-E(\mathbf{v}, \mathbf{h}; \theta)\big)$$

Markov random fields, Boltzmann machines, log-linear models.

Restricted Boltzmann Machines



Product of Experts formulation.

The joint distribution is given by:

$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\Big(\sum_{ij} W_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j\Big)$$

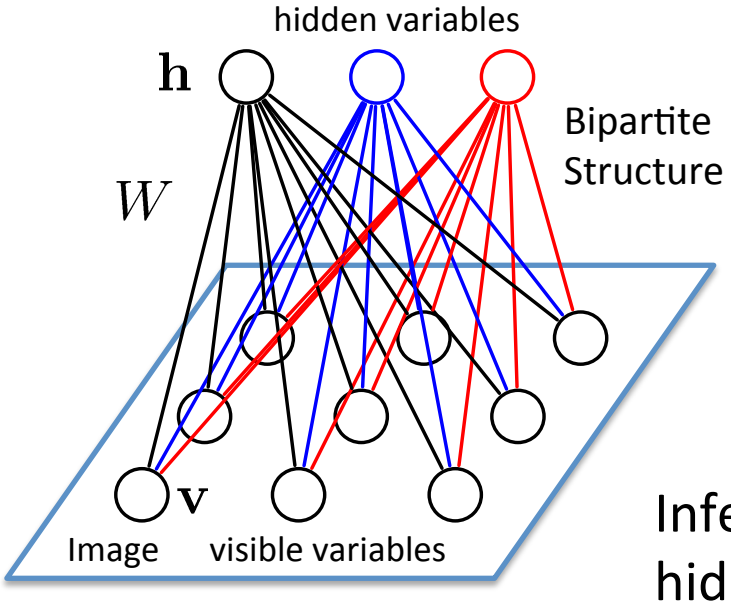
where the undirected edges in the graphical model represent $\{W_{ij}\}$.

Marginalizing over the states of hidden variables:

$$P_\theta(\mathbf{v}) = \sum_{\mathbf{h}} P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \prod_i \exp(b_i v_i) \prod_j \underbrace{\Big(1 + \exp(a_j + \sum_i W_{ij} v_i)\Big)}_{\text{Product of experts}}$$

Markov random fields, Boltzmann machines, log-linear models.

Restricted Boltzmann Machines



Restricted: No interaction between hidden variables

Inferring the distribution over the hidden variables is easy:

$$P(\mathbf{h}|\mathbf{v}) = \prod_j P(h_j|\mathbf{v}) \quad P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}$$

Factorizes: Easy to compute

Similarly:

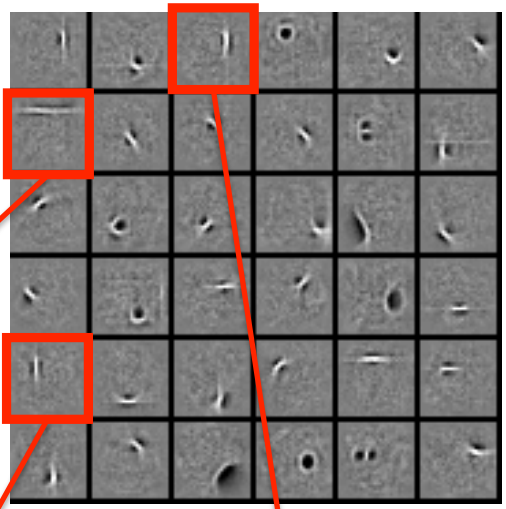
$$P(\mathbf{v}|\mathbf{h}) = \prod_i P(v_i|\mathbf{h}) \quad P(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)}$$

Markov random fields, Boltzmann machines, log-linear models.

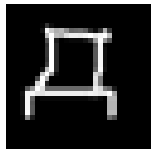
Learning Features

Observed Data
Subset of 25,000 characters

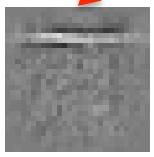
Learned W: "edges"
Subset of 1000 features



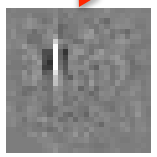
New Image:



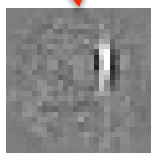
$$= \sigma\left(0.99 \times \right.$$



$$+ \quad 0.97 \times$$



$$+ \quad 0.82 \times$$



$$\left. \cdots \right)$$

$p(h_7 = 1|v)$

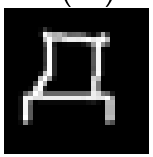
$p(h_{29} = 1|v)$

Most hidden variables are off

$\sigma(x) = \frac{1}{1+\exp(-x)}$

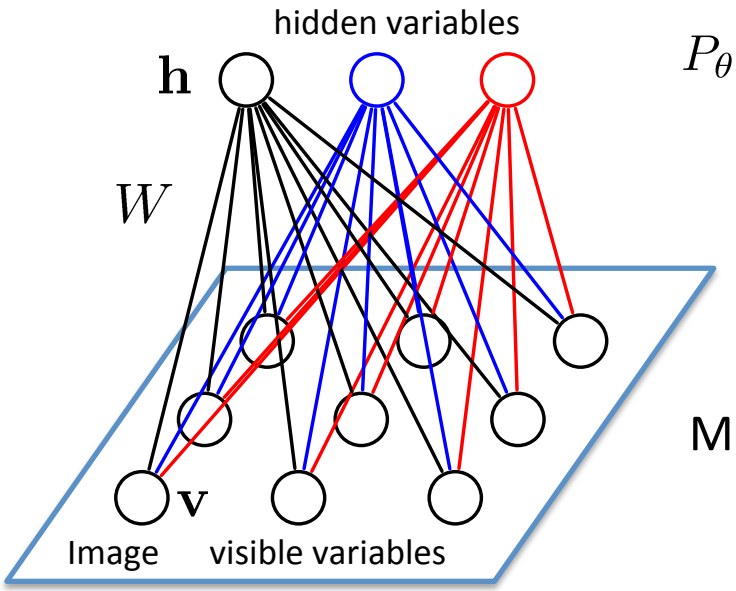
Logistic Function: Suitable for modeling binary images

Represent:



as $P(\mathbf{h}|\mathbf{v}) = [0,\ 0,\ 0.82,\ 0,\ 0,\ 0.99,\ 0,\ 0\ \ldots\ ]$

Model Learning



$$P_\theta(\mathbf{v}) = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}} \exp\left[\mathbf{v}^\top W \mathbf{h} + \mathbf{a}^\top \mathbf{h} + \mathbf{b}^\top \mathbf{v}\right]$$

Given a set of *i.i.d.* training examples $\mathcal{D} = \{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, ..., \mathbf{v}^{(N)}\}$, we want to learn model parameters $\theta = \{W, a, b\}$.

Maximize (penalized) log-likelihood objective:

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log P_\theta(\mathbf{v}^{(n)}) - \underbrace{\frac{\lambda}{N} ||W||_F^2}_{\text{Regularization}}$$

Derivative of the log-likelihood:

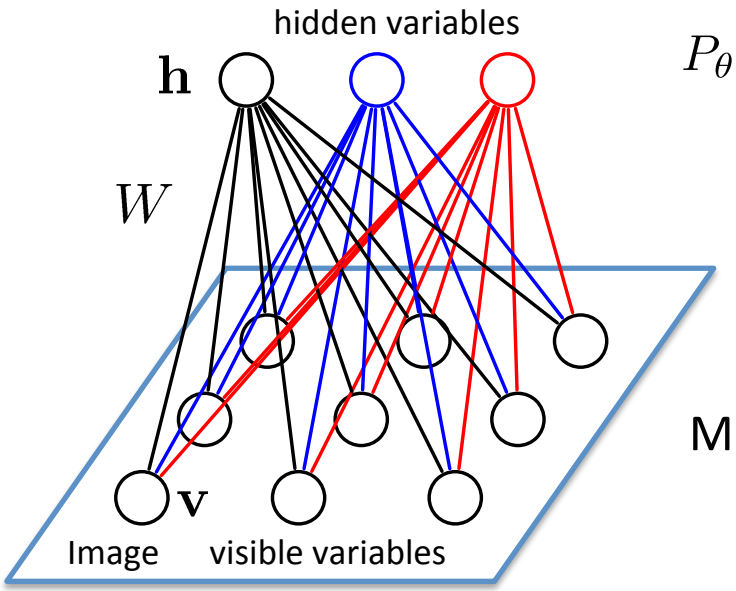
$$\frac{\partial L(\theta)}{\partial W_{ij}} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial}{\partial W_{ij}} \log \left(\sum_{\mathbf{h}} \exp\left[\mathbf{v}^{(n)\top} W \mathbf{h} + \mathbf{a}^\top \mathbf{h} + \mathbf{b}^\top \mathbf{v}^{(n)}\right] \right) - \frac{\partial}{\partial W_{ij}} \log \mathcal{Z}(\theta) - \frac{2\lambda}{N} W_{ij}$$

$$= \mathrm{E}_{P_{data}}[v_i h_j] - \underbrace{\mathrm{E}_{P_\theta}[v_i h_j]}_{\text{Difficult to compute: exponentially many configurations}} - \frac{2\lambda}{N} W_{ij}$$

$$P_{data}(\mathbf{v}, \mathbf{h}; \theta) = P(\mathbf{h}|\mathbf{v}; \theta) P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N} \sum_{n} \delta(\mathbf{v} - \mathbf{v}^{(n)})$$

Model Learning



$$P_\theta(\mathbf{v}) = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}} \exp\left[\mathbf{v}^\top W \mathbf{h} + \mathbf{a}^\top \mathbf{h} + \mathbf{b}^\top \mathbf{v}\right]$$

Given a set of *i.i.d.* training examples $\mathcal{D} = \{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, ..., \mathbf{v}^{(N)}\}$, we want to learn model parameters $\theta = \{W, a, b\}$.

Maximize (penalized) log-likelihood objective:

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log P_\theta(\mathbf{v}^{(n)}) - \frac{\lambda}{N} ||W||_F^2$$

Derivative of the log-likelihood:

$$\frac{\partial L(\theta)}{\partial W_{ij}} = \mathrm{E}_{P_{data}}[v_i h_j] - \mathrm{E}_{P_\theta}[v_i h_j] - \frac{2\lambda}{N} W_{ij}$$

Approximate maximum likelihood learning:

Contrastive Divergence (Hinton 2000)
MCMC-MLE estimator (Geyer 1991)

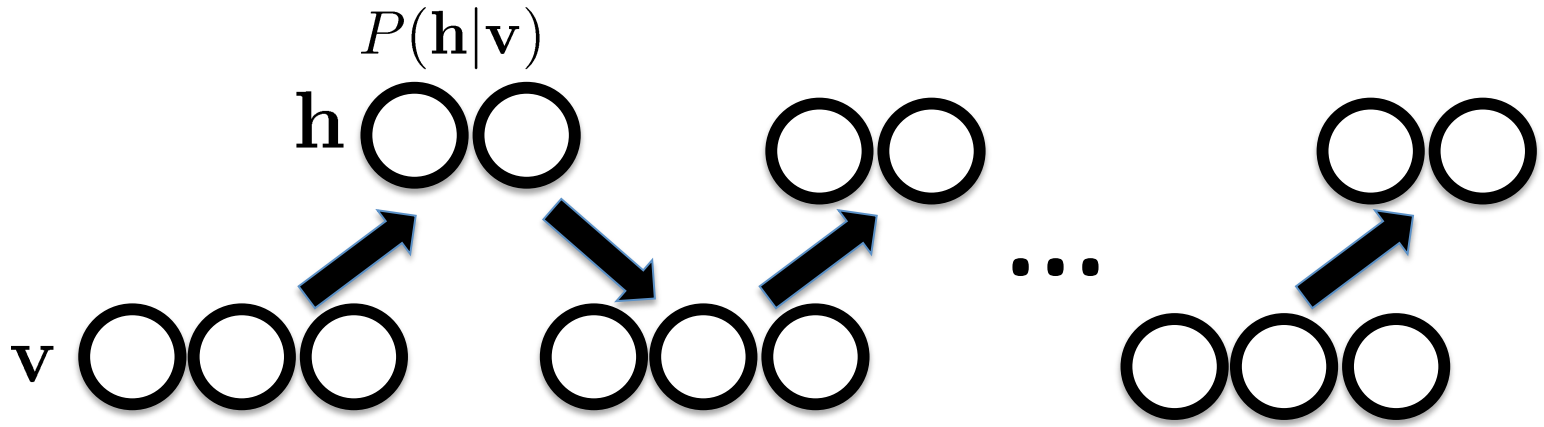
Tempered MCMC
(Salakhutdinov, NIPS 2009)

Pseudo Likelihood (Besag 1977)
Composite Likelihoods (Lindsay, 1988; Varin 2008)

Adaptive MCMC
(Salakhutdinov, ICML 2010)

Contrastive Divergence

Run Markov chain for a few steps (e.g. one step):



$$P(\mathbf{h}|\mathbf{v}) = \prod_j P(h_j|\mathbf{v}) \quad P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}$$

$$P(\mathbf{v}|\mathbf{h}) = \prod_i P(v_i|\mathbf{h}) \quad P(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)}$$

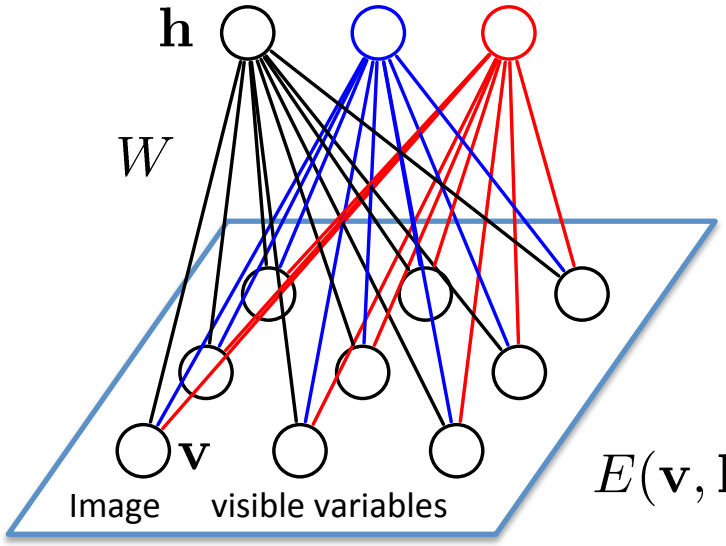
Update model parameters:

$$\Delta W_{ij} = \mathrm{E}_{P_{data}}[v_i h_j] - \mathrm{E}_{P_1}[v_i h_j]$$

Hinton, Neural Computation 2002

RBMs for Images

Gaussian-Bernoulli RBM:



$$P_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp(-E(\mathbf{v}, \mathbf{h}; \theta))$$

Define energy functions for various data modalities:

$$E(\mathbf{v}, \mathbf{h}; \theta) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{ij} W_{ij} h_j \frac{v_i}{\sigma_i} - \sum_j a_j h_j$$

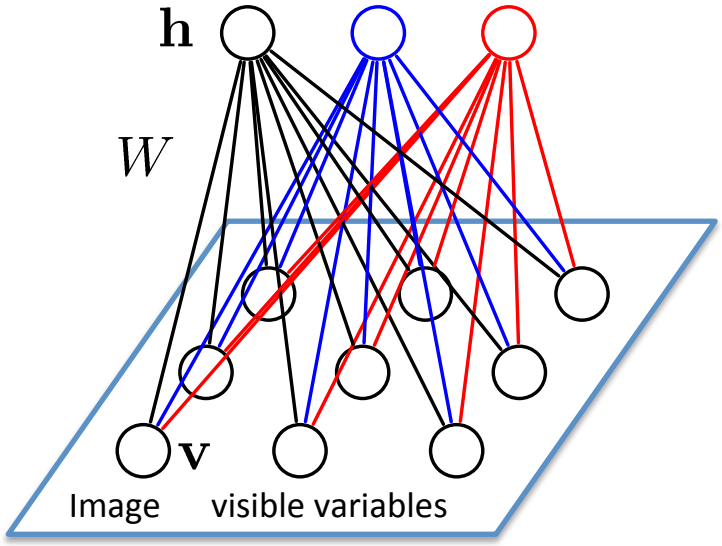
$$P(v_i = x|\mathbf{h}) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x - b_i - \sigma_i \sum_j W_{ij} h_j)^2}{2\sigma_i^2}\right)$$ Gaussian

$$P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} \frac{v_i}{\sigma_i} - a_j)}$$ Bernoulli

(Salakhutdinov & Hinton, NIPS 2007)

RBMs for Images

Gaussian-Bernoulli RBM:



Interpretation: Mixture of exponential number of Gaussians

$$P_\theta(\mathbf{v}) = \sum_{\mathbf{h}} P_\theta(\mathbf{v}|\mathbf{h}) P_\theta(\mathbf{h}),$$

where

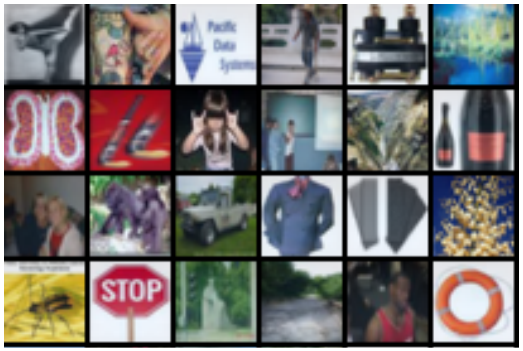
$$P_\theta(\mathbf{h}) = \int_{\mathbf{v}} P_\theta(\mathbf{v}, \mathbf{h}) d\mathbf{v}$$ is an implicit prior, and

$$P(v_i = x|\mathbf{h}) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x - b_i - \sigma_i \sum_j W_{ij} h_j)^2}{2\sigma_i^2}\right)$$ Gaussian

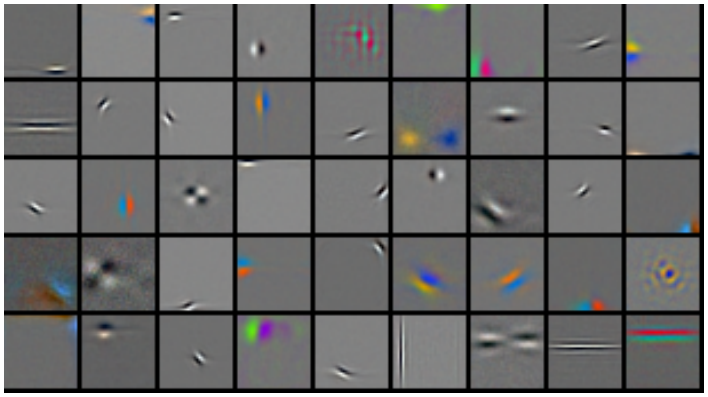
RBMs for Images and Text

Images: Gaussian-Bernoulli RBM

4 million **unlabelled** images



Learned features (out of 10,000)



Text: Multinomial-Bernoulli RBM

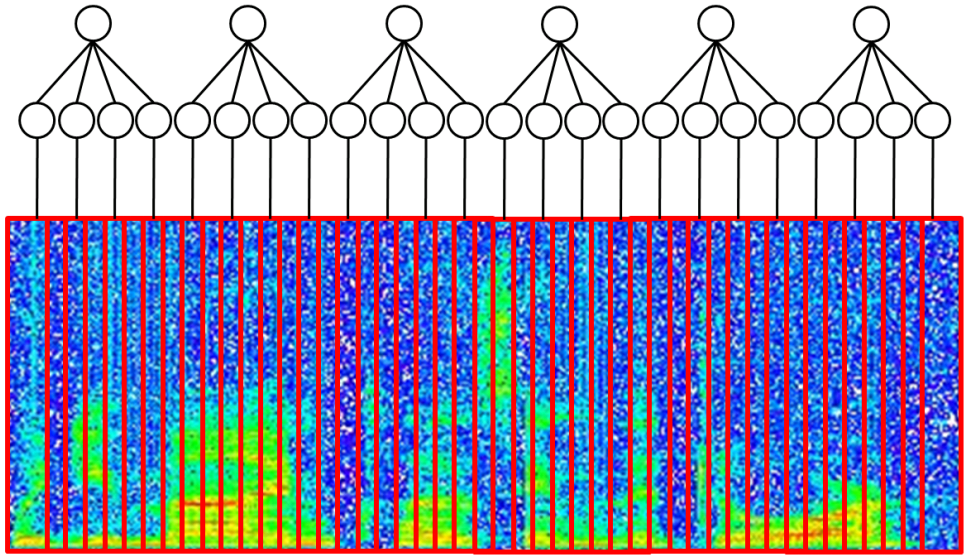
Reuters dataset:
804,414 **unlabeled**
newswire stories
Bag-of-Words

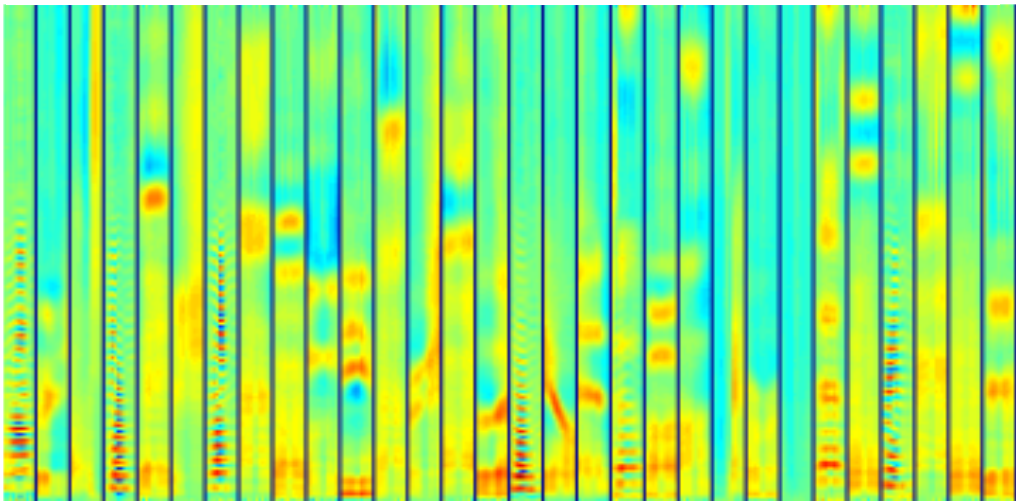
Learned features: ``topics''

russian	clinton	computer	trade	stock
russia	house	system	country	wall
moscow	president	product	import	street
yeltsin	bill	software	world	point
soviet	congress	develop	economy	dow

(Salakhutdinov & Hinton SIGIR 2007, NIPS 2010)

Speech

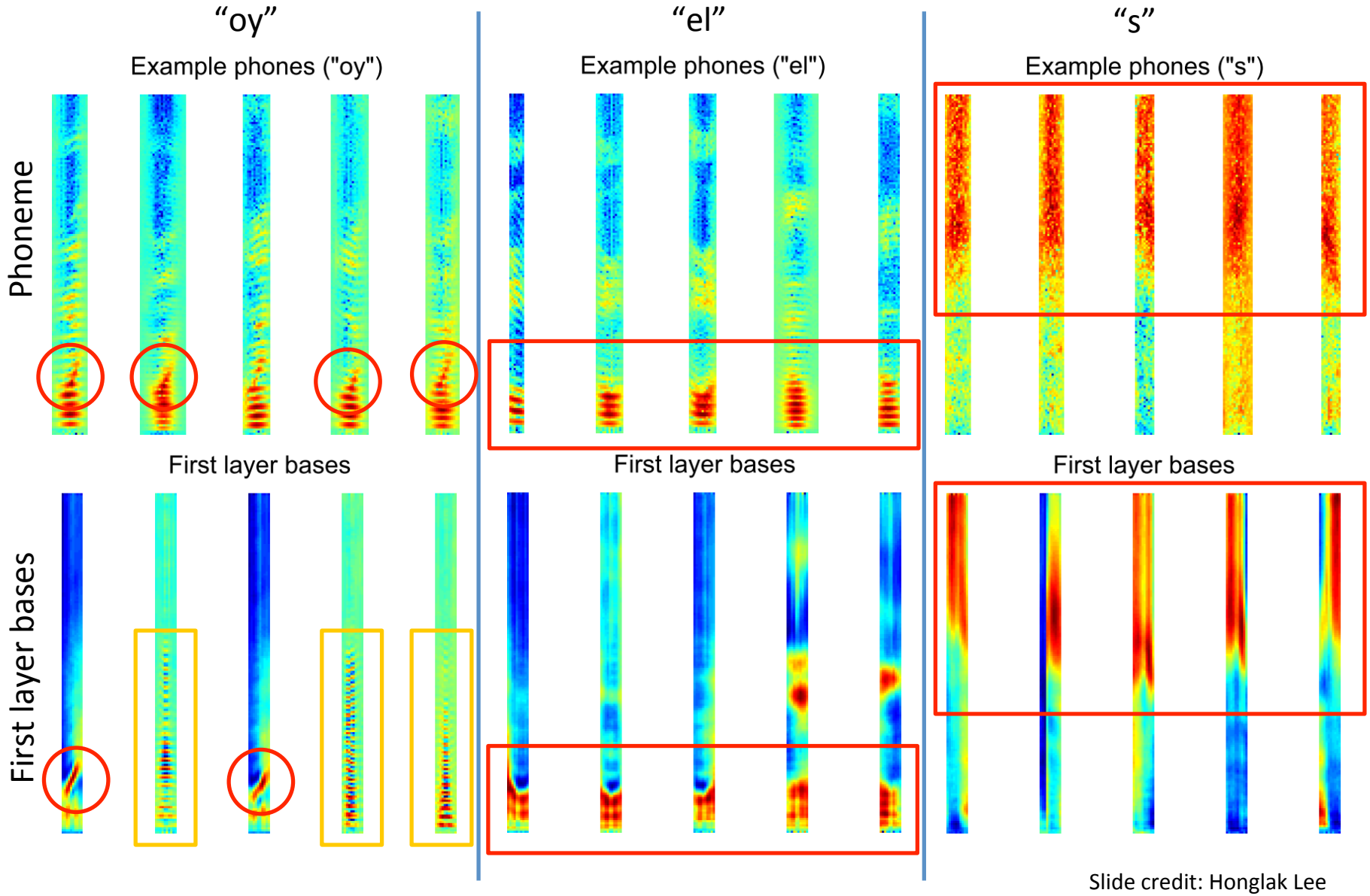




Learned first-layer bases

Lee et.al., NIPS 2009

Comparison of bases to phonemes

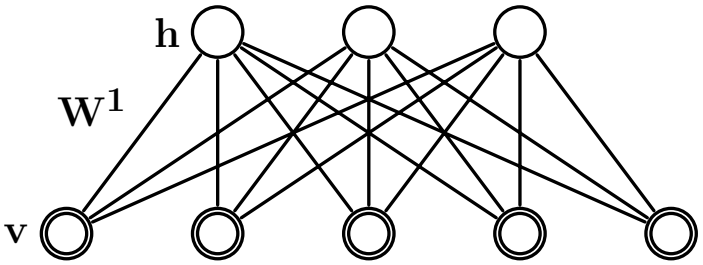


Slide credit: Honglak Lee

Collaborative Filtering

$$P_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(\sum_{ijk} W_{ij}^k v_i^k h_j + \sum_{ik} b_i^k v_i^k + \sum_j a_j h_j \right)$$

Bernoulli hidden: user preferences



Multinomial visible: user ratings

Netflix dataset:
480,189 users
17,770 movies
Over 100 million ratings

Learned features: ``genre''

Fahrenheit 9/11	Independence Day
Bowling for Columbine	The Day After Tomorrow
The People vs. Larry Flynt	Con Air
Canadian Bacon	Men in Black II
La Dolce Vita	Men in Black
Friday the 13th	Scary Movie
The Texas Chainsaw Massacre	Naked Gun
Children of the Corn	Hot Shots!
Child's Play	American Pie
The Return of Michael Myers	Police Academy

State-of-the-art performance on the Netflix dataset.

Relates to **Probabilistic Matrix Factorization**

(Salakhutdinov & Mnih ICML 2007)

Multiple Application Domains

- Natural Images
- Text/Documents
- Collaborative Filtering / Matrix Factorization
- Video (Langford, Salakhutdinov and Zhang, ICML 2009)
- Motion Capture (Taylor et.al. NIPS 2007)
- Speech Perception (Dahl et. al. NIPS 2010, Lee et.al. NIPS 2010)

Same learning algorithm --
multiple input domains.

Limitations on the types of structure that can be represented by a single layer of low-level features!

Talk Roadmap

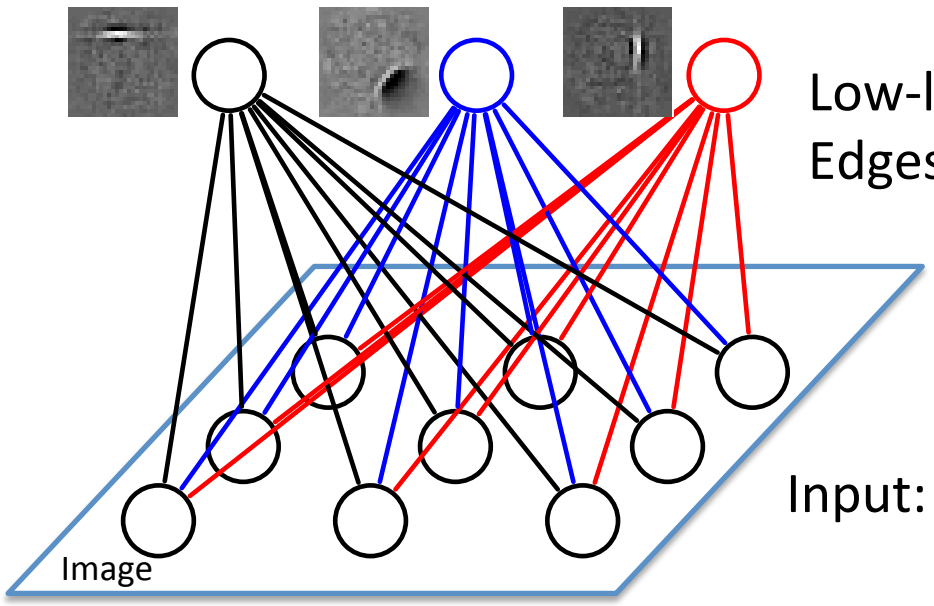
Part 1: Deep Networks

- Introduction, Graphical Models.
- Restricted Boltzmann Machines: Learning low-level features.
- Deep Belief Networks: Learning Part-based Hierarchies.

Part 2: Deep Boltzmann Machines.

- Inference and Learning
- Advanced Deep Models

Deep Belief Network



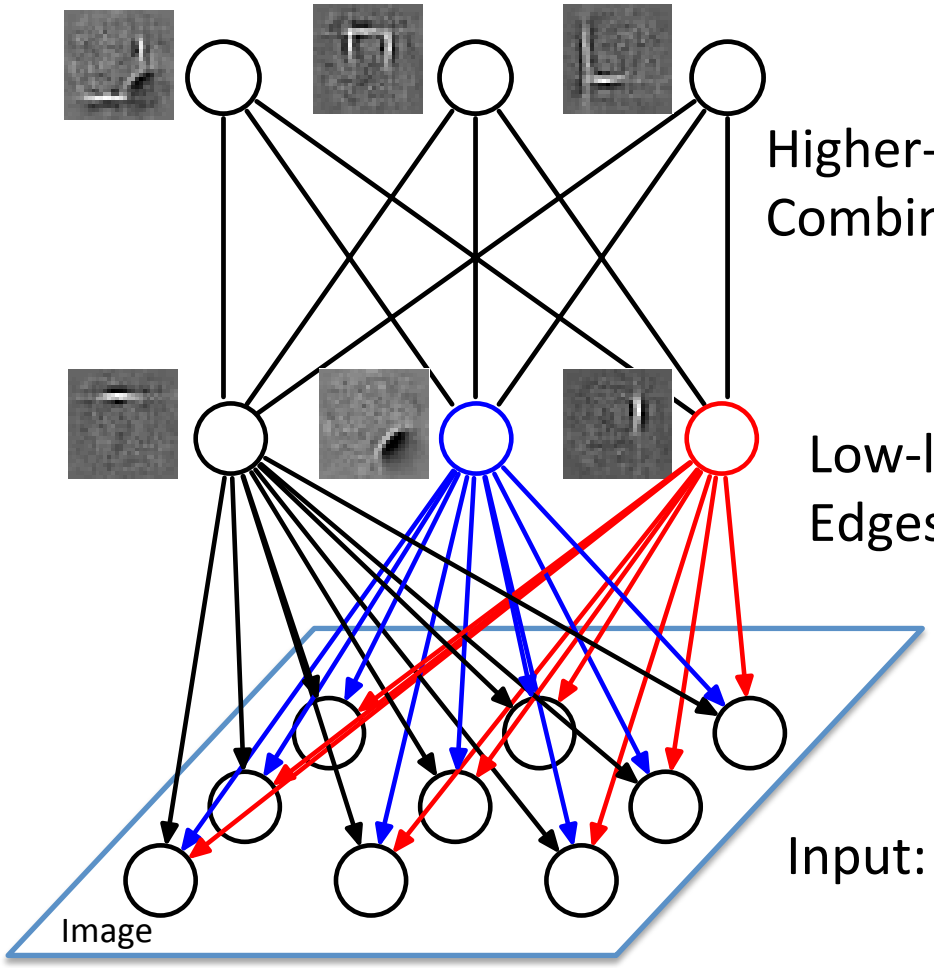
Low-level features:
Edges

Built from **unlabeled** inputs.

Input: Pixels

(Hinton et.al. Neural Computation 2006)

Deep Belief Network



Internal representations capture higher-order statistical structure

Higher-level features:
Combination of edges

Low-level features:
Edges

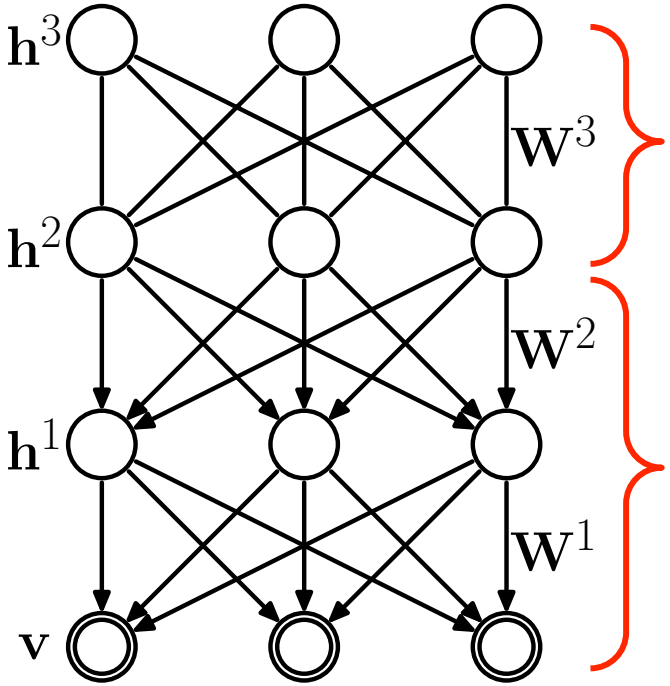
Built from **unlabeled** inputs.

Input: Pixels

(Hinton et.al. Neural Computation 2006)

Deep Belief Network

Deep Belief Network



RBM

Sigmoid Belief Network

The joint probability distribution factorizes:

$$P(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3) = P(\mathbf{v}|\mathbf{h}^1)P(\mathbf{h}^1|\mathbf{h}^2)P(\mathbf{h}^2, \mathbf{h}^3)$$

Sigmoid Belief Network

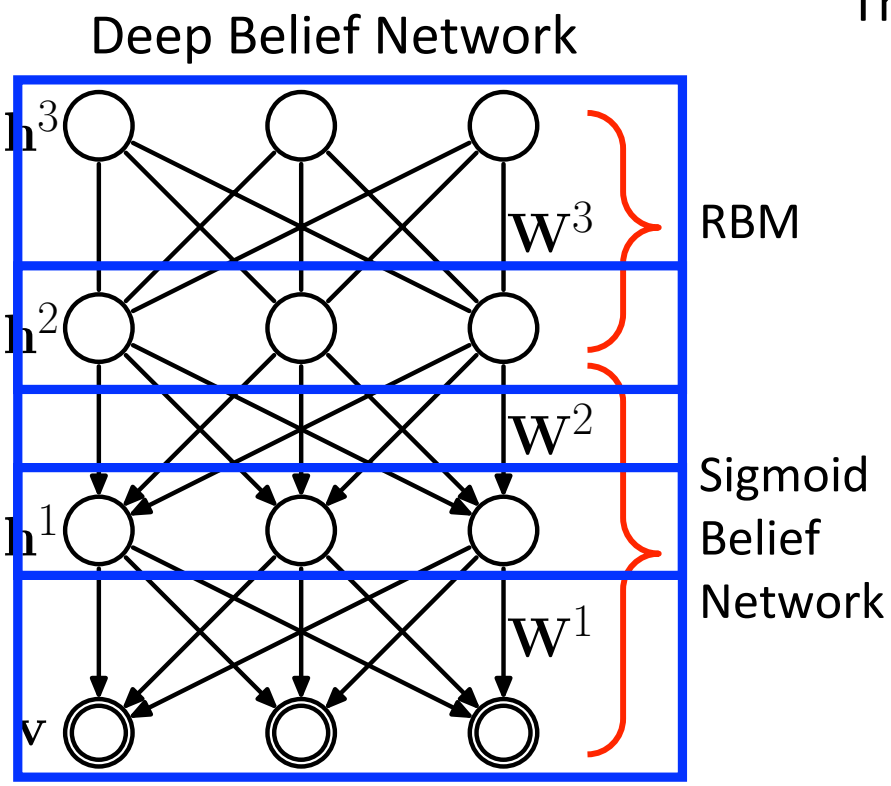
RBM

$$P(\mathbf{h}^2, \mathbf{h}^3) = \frac{1}{\mathcal{Z}(W^3)} \exp\left[\mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$

$$P(\mathbf{h}^1|\mathbf{h}^2) = \prod_j P(h_j^1|\mathbf{h}^2) \qquad P(h_j^1 = 1|\mathbf{h}^2) = \frac{1}{1 + \exp\left(-\sum_k W_{jk}^2 h_k^2\right)}$$

$$P(\mathbf{v}|\mathbf{h}^1) = \prod_i P(v_i|\mathbf{h}^1) \qquad P(v_i = 1|\mathbf{h}^1) = \frac{1}{1 + \exp\left(-\sum_j W_{ij}^1 h_j^1\right)}$$

Deep Belief Network



Unsupervised Feature Learning.

The joint probability distribution factorizes:

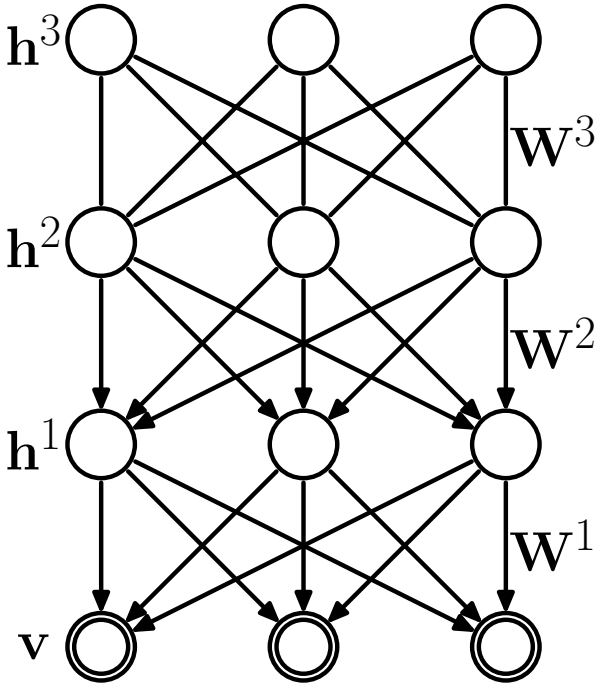
$$P(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3) = P(\mathbf{v}|\mathbf{h}^1)P(\mathbf{h}^1|\mathbf{h}^2)P(\mathbf{h}^2, \mathbf{h}^3)$$

Layerwise Pretraining:

- Learn and freeze 1st layer RBM
- Treat inferred values $P(\mathbf{h}^1|\mathbf{v})$ as the data for training 2nd-layer RBM.
- Learn and freeze 2nd layer RBM.
- Proceed to the next layer.

Layerwise Pretraining

Deep Belief Network



Efficient layer-wise pretraining algorithm.

$$\log P_\theta(\mathbf{v}) = \sum_{\mathbf{h}^1} P_\theta(\mathbf{v}, \mathbf{h}^1) \geq \underbrace{\sum_{\mathbf{h}^1} Q_\phi(\mathbf{h}^1|\mathbf{v}) \log \frac{P_\theta(\mathbf{h}^1, \mathbf{v})}{Q_\phi(\mathbf{h}^1|\mathbf{v})}}_{\text{Variational Lower Bound}}$$

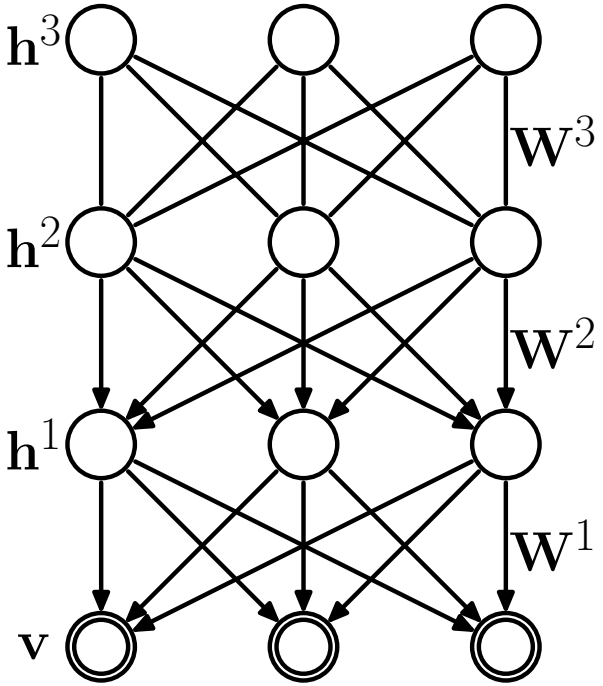
$$= \sum_{\mathbf{h}^1} \underbrace{Q_\phi(\mathbf{h}^1|\mathbf{v}) \left[\log P_\theta(\mathbf{v}|\mathbf{h}^1; W^1) \right]}_{\text{Likelihood term}} + \underbrace{\mathcal{H}(Q_\phi(\mathbf{h}^1|\mathbf{v}))}_{\text{Entropy functional}}$$

$$+ \sum_{\mathbf{h}^1} Q_\phi(\mathbf{h}^1|\mathbf{v}) \underbrace{\log P_\theta(\mathbf{h}^1; W^2)}_{\text{Replace with a second layer RBM}}$$

Similar arguments for pretraining a Deep Boltzmann machine

Layerwise Pretraining

Deep Belief Network



Efficient layer-wise pretraining algorithm.

$$\log P_\theta(\mathbf{v}) = \sum_{\mathbf{h}^1} P_\theta(\mathbf{v}, \mathbf{h}^1) \geq \sum_{\mathbf{h}^1} Q_\phi(\mathbf{h}^1|\mathbf{v}) \log \frac{P_\theta(\mathbf{h}^1, \mathbf{v})}{Q_\phi(\mathbf{h}^1|\mathbf{v})}$$

Variational Lower Bound

$$= \sum_{\mathbf{h}^1} \ldots \mathbf{v}))$$

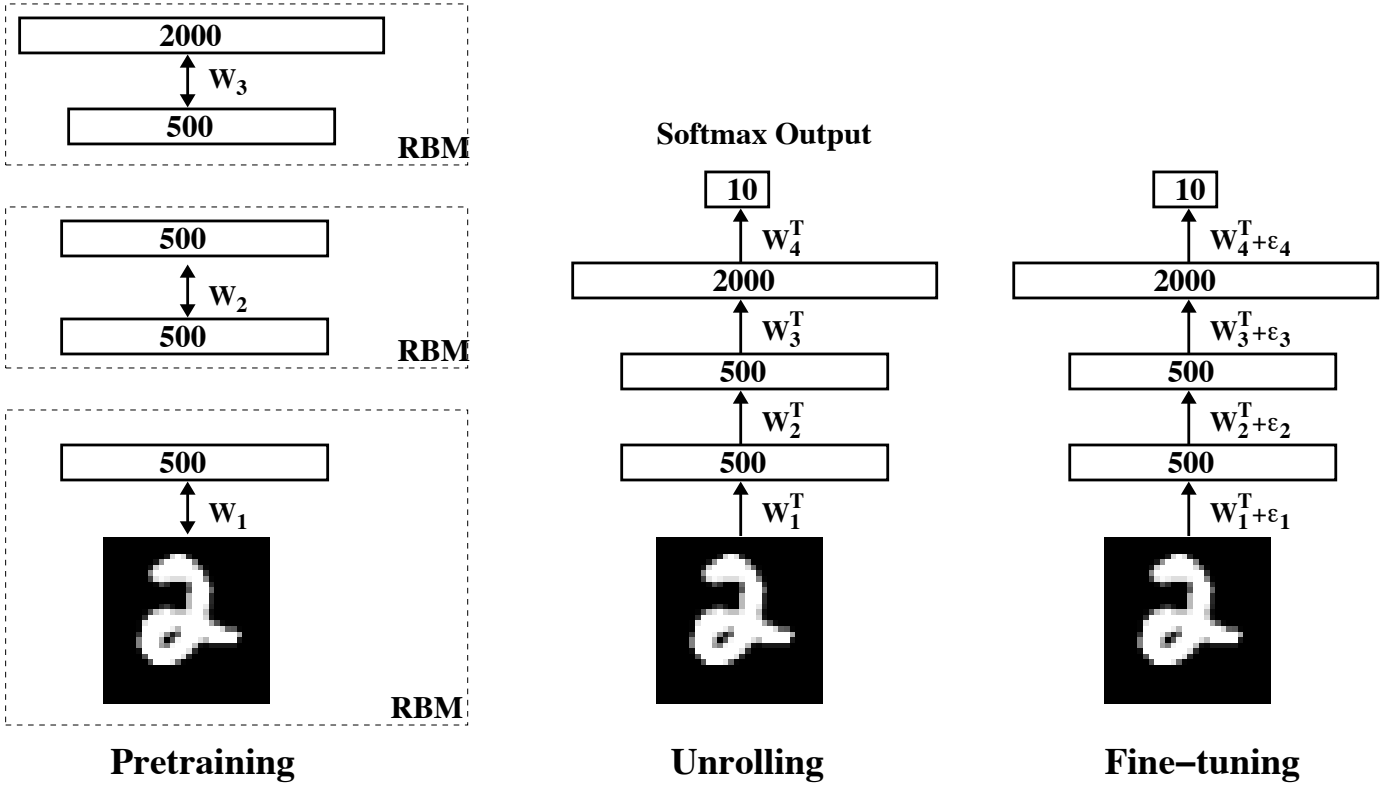
ional

Layerwise pretraining improves variational lower bound

Replace with a second layer RBM

Similar arguments for pretraining a Deep Boltzmann machine

DBNs for Classification



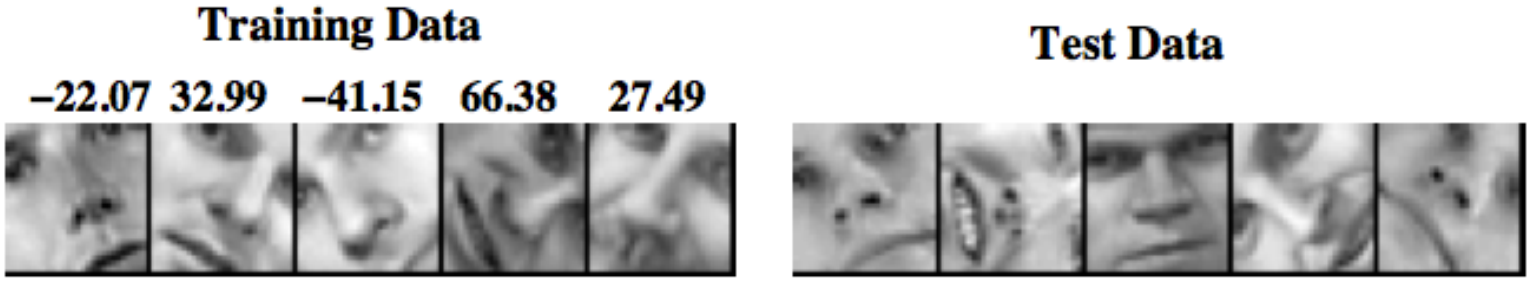
• After layer-by-layer **unsupervised pretraining**, discriminative fine-tuning by backpropagation achieves an error rate of 1.2% on MNIST. SVM's get 1.4% and randomly initialized backprop gets 1.6%.

• Clearly unsupervised learning helps generalization. It ensures that most of the information in the weights comes from modeling the input data.

(Hinton and Salakhutdinov, Science 2006)

DBNs for Regression

Predicting the orientation of a face patch



Training Data: 1000 face patches of 30 training people.

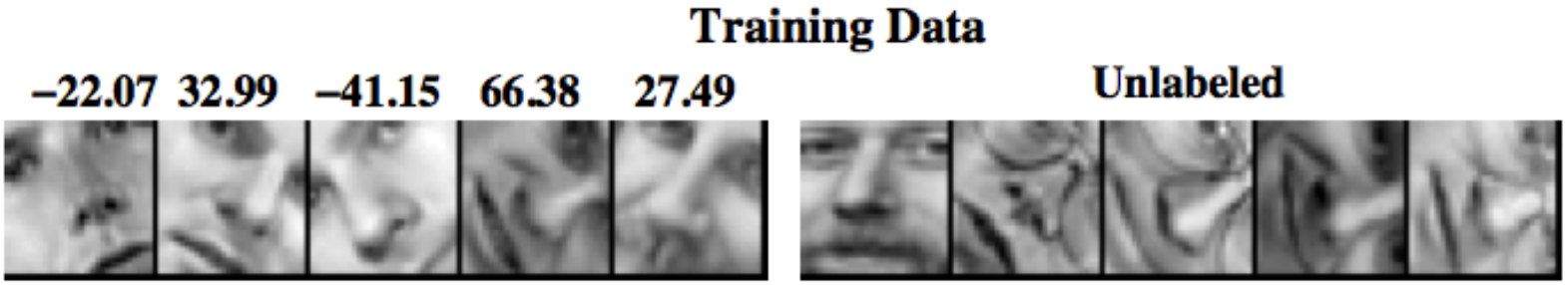
Test Data: 1000 face patches of **10 new people**.

Regression Task: predict orientation of a new face.

Gaussian Processes with spherical Gaussian kernel achieves a RMSE (root mean squared error) of 16.33 degree.

(Salakhutdinov and Hinton, NIPS 2007)

DBNs for Regression

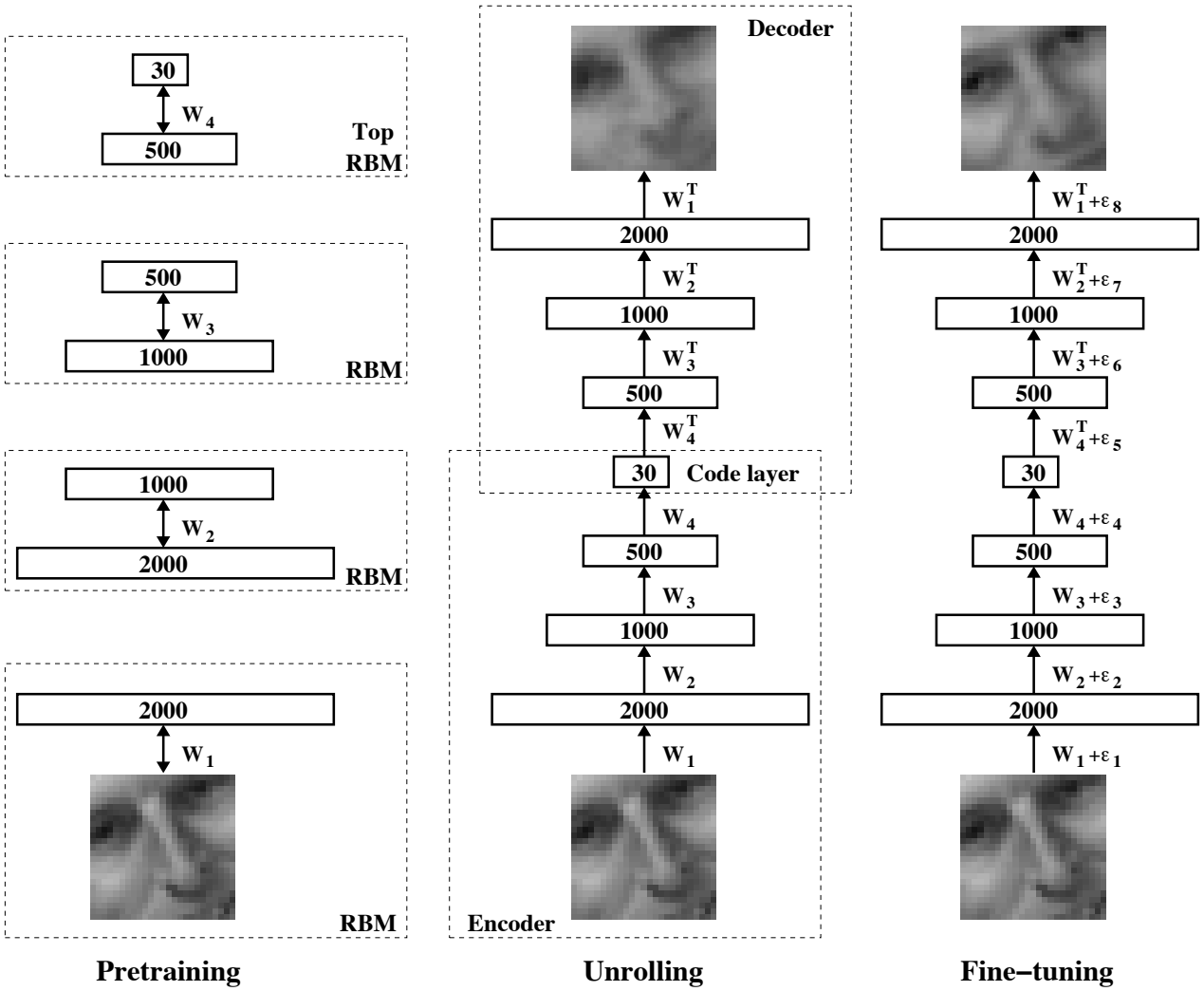


Additional Unlabeled Training Data: 12000 face patches from 30 training people.

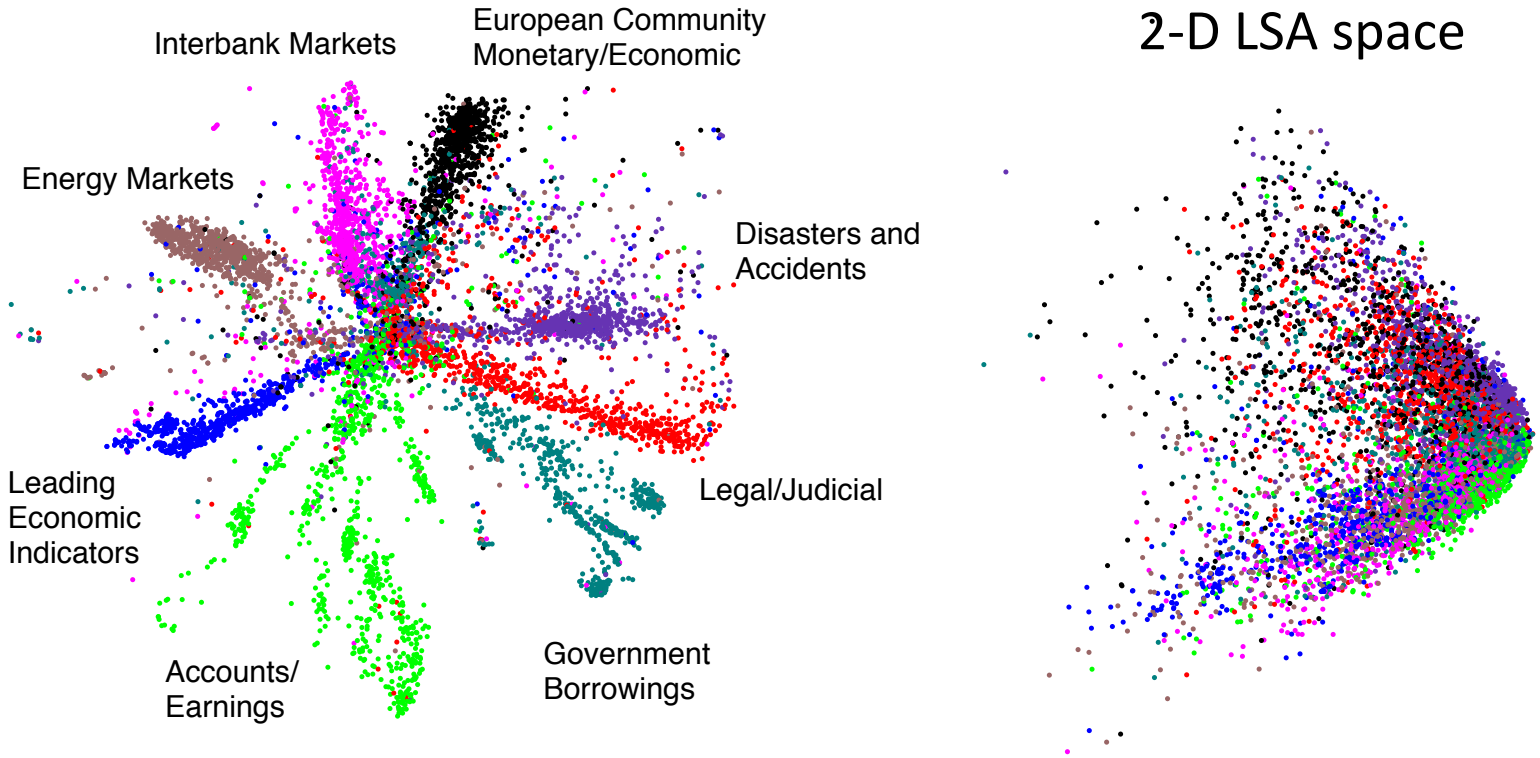
- Pretrain a stack of RBMs: 784-1000-1000-1000.
- **Features were extracted with no idea of the final task.**

The same GP on the top-level features:	RMSE: 11.22
GP with fine-tuned covariance Gaussian kernel:	RMSE: 6.42
Standard GP without using DBNs:	RMSE: 16.33

Deep Autoencoders



Information Retrieval

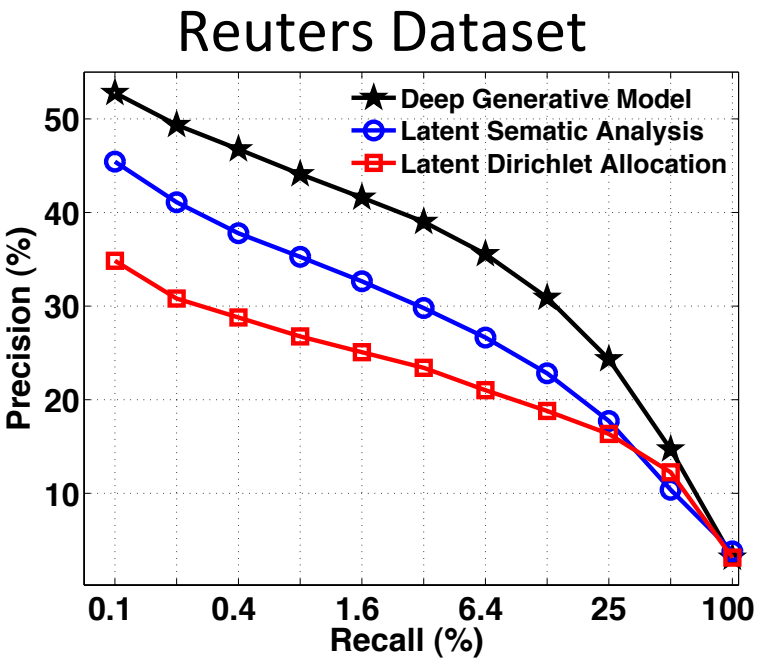


• The Reuters Corpus Volume II contains 804,414 newswire stories (randomly split into **402,207 training** and **402,207 test).**

• "Bag-of-words" representation: each article is represented as a vector containing the counts of the most frequently used 2000 words in the training set.

(Hinton and Salakhutdinov, Science 2006)

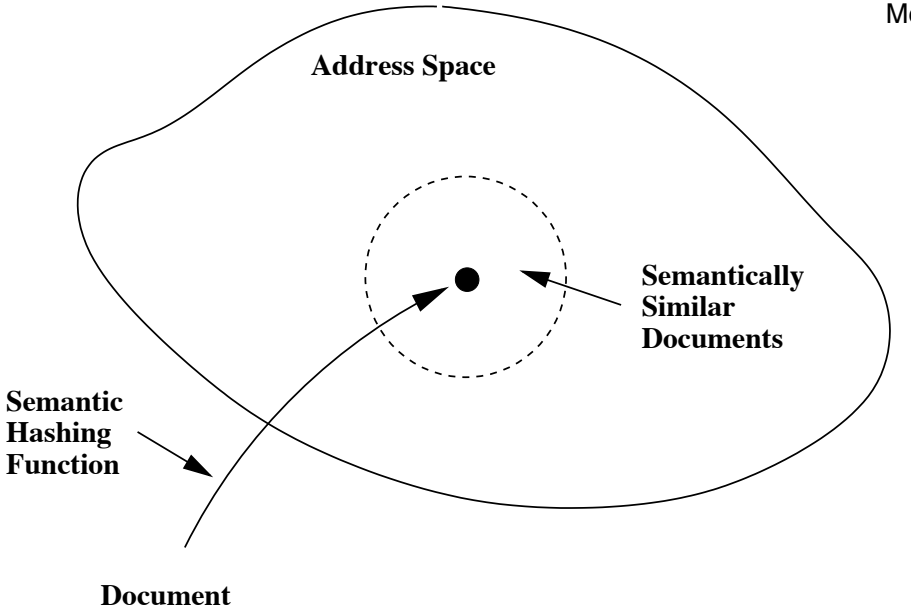
Information Retrieval

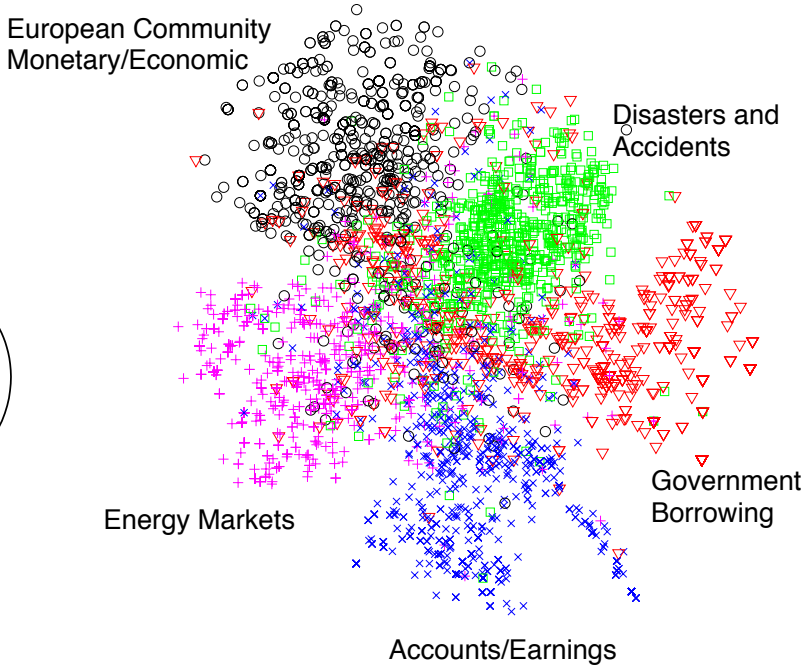


Reuters dataset: 804,414 newswire stories.

Deep generative model significantly outperforms LSA and LDA topic models

Semantic Hashing



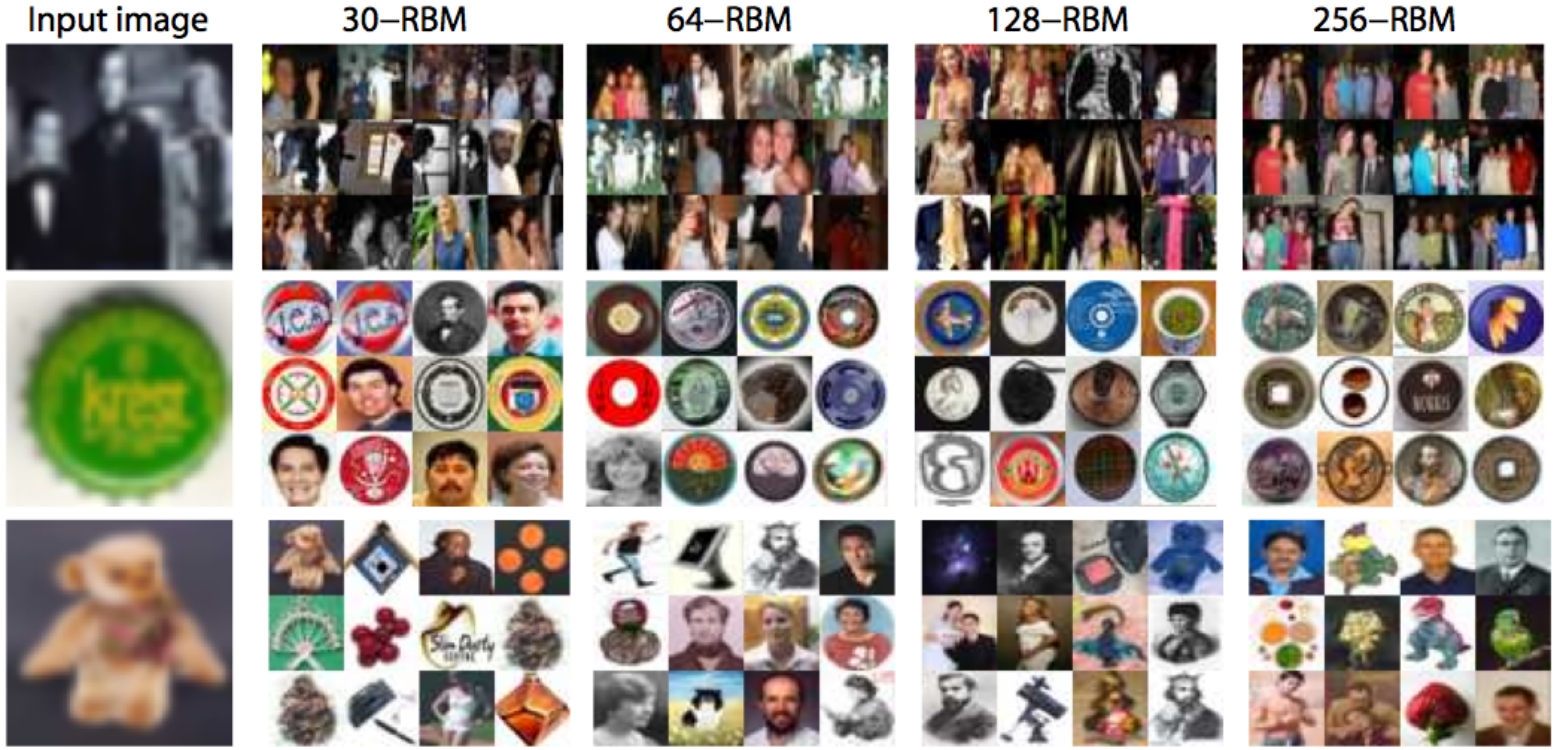


- Learn to map documents into **semantic 20-D binary codes.**
- Retrieve similar documents stored at the nearby addresses **with no search at all.**

(Salakhutdinov and Hinton, SIGIR 2007)

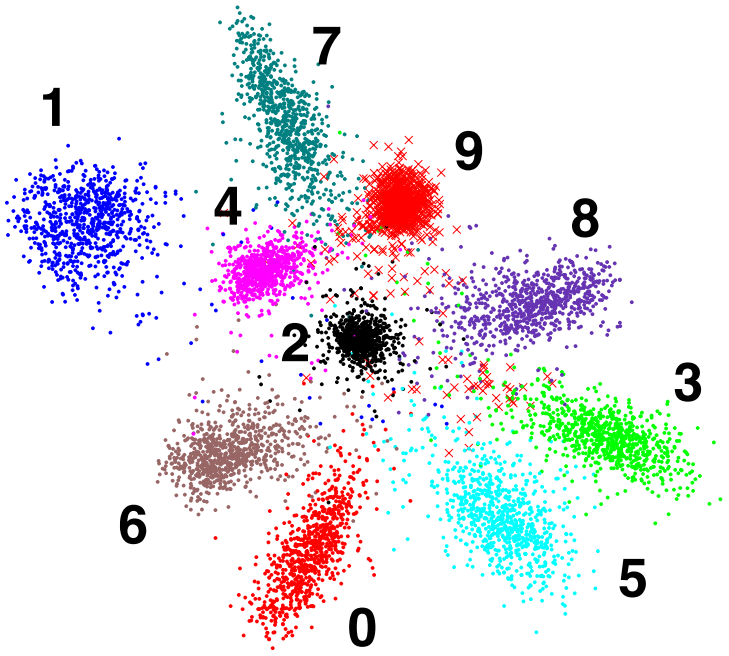
Searching Large Image Database using Binary Codes

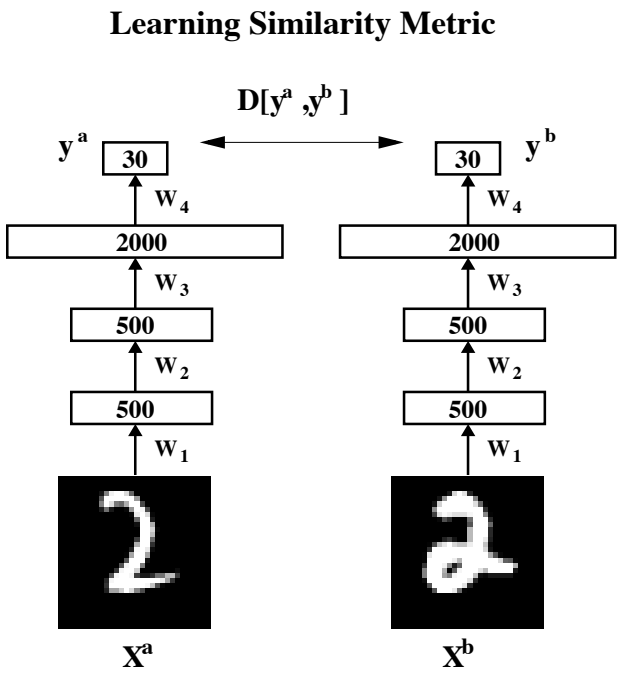
- Map images into binary codes for fast retrieval.



- Small Codes, Torralba, Fergus, Weiss, CVPR 2008
- Spectral Hashing, Y. Weiss, A. Torralba, R. Fergus, NIPS 2008
- Kulis and Darrell, NIPS 2009, Gong and Lazebnik, CVPR 20111
- Norouzi and Fleet, ICML 2011,

Learning Similarity Measures

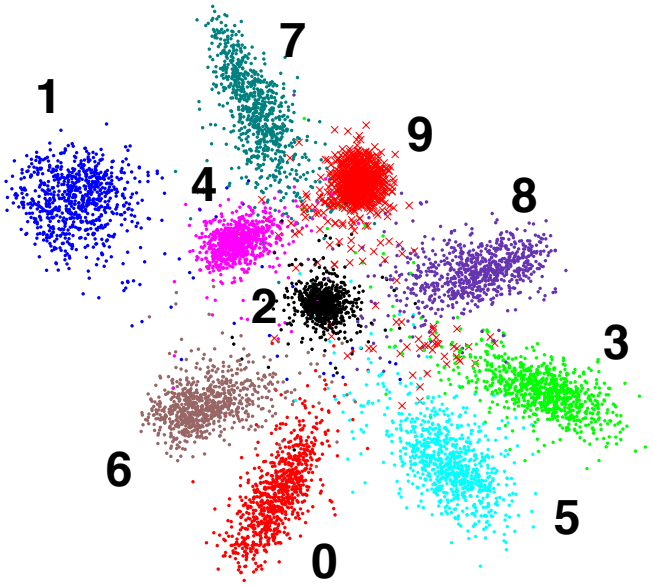


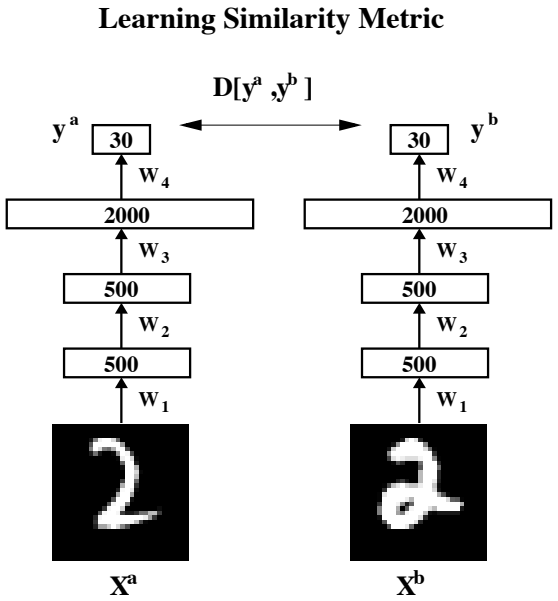


- Learn a nonlinear transformation of the input space.
- Optimize to make KNN perform well in the low-dimensional feature space

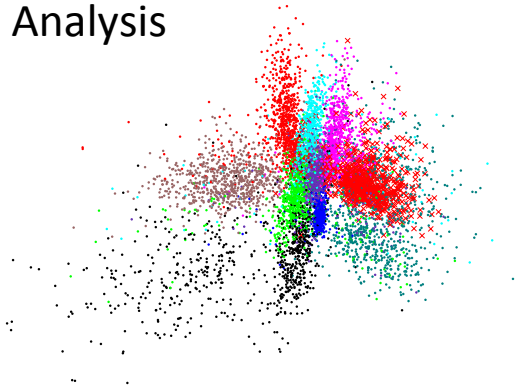
(Salakhutdinov and Hinton, AI and Statistics 2007)

Compare to Other Approaches

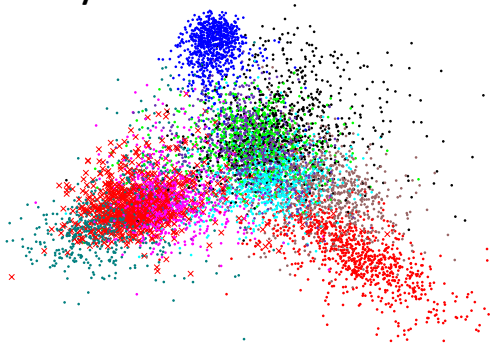




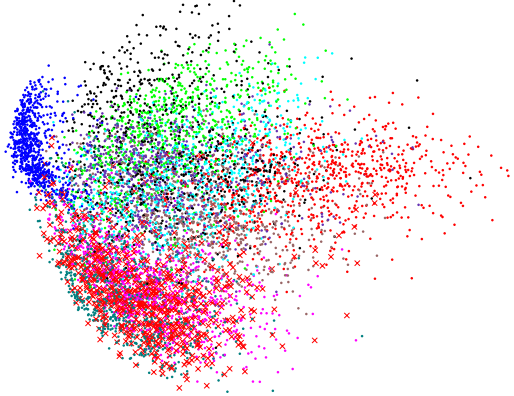
Neighborhood Component Analysis



Linear discriminant Analysis



PCA



Talk Roadmap

Part 1: Deep Networks

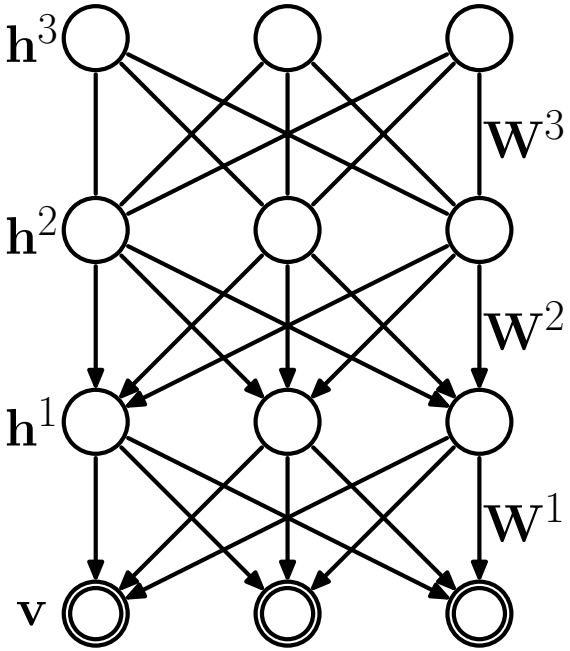
- Introduction, Graphical Models.
- Restricted Boltzmann Machines: Learning low-level features.
- Deep Belief Networks: Learning Part-based Hierarchies.

Part 2: Deep Boltzmann Machines.

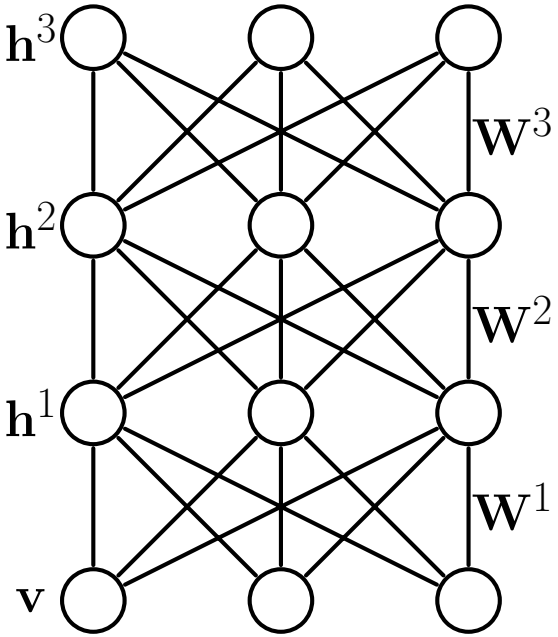
- Inference and Learning
- Advanced Deep Models

DBNs vs. DBMs

Deep Belief Network



Deep Boltzmann Machine



DBNs are hybrid models:

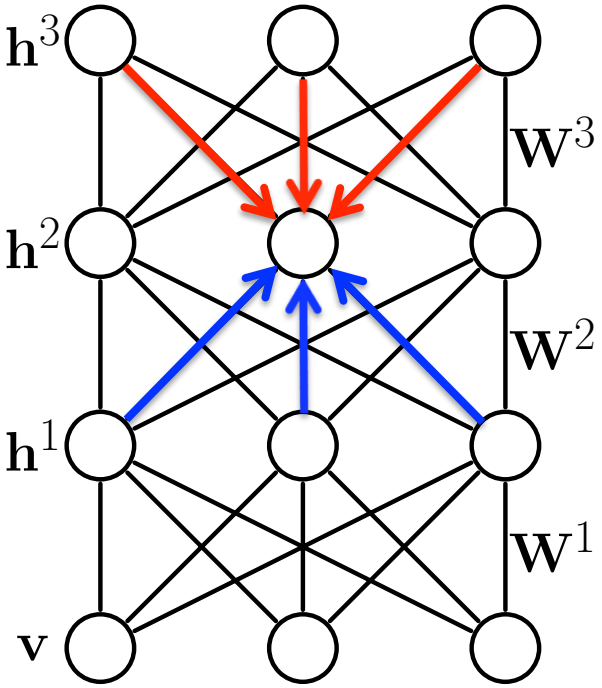
- Inference in DBNs is problematic due to **explaining away**.
- Only greedy pretrainig, **no joint optimization over all layers**.
- Approximate inference is feed-forward: **no bottom-up and top-down**.

Introduce a new class of models called Deep Boltzmann Machines.

Mathematical Formulation

$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \mathbf{h}^{1\top} W^2 \mathbf{h}^2 + \mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$

Deep Boltzmann Machine



Input

$\theta = \{W^1, W^2, W^3\}$ model parameters

- Dependencies between hidden variables.
- All connections are undirected.
- Bottom-up and Top-down:

$$P(h_k^2 = 1 | \mathbf{h}^1, \mathbf{h}^3) = \sigma\left(\sum_j W_{jk}^2 h_j^1 + \sum_m W_{km}^3 h_m^3\right)$$

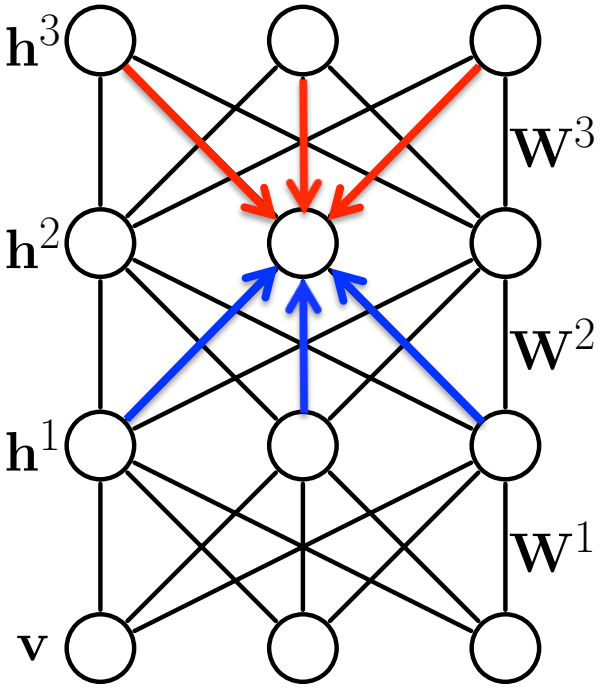
Bottom-up Top-Down

Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio et.al.), Deep Belief Nets (Hinton et.al.)

Mathematical Formulation

$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \mathbf{h}^{1\top} W^2 \mathbf{h}^2 + \mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$

Deep Boltzmann Machine



Input

- Conditional Distributions:

$$P(h_j^1 = 1|\mathbf{v}, \mathbf{h}^2) = \sigma\left(\sum_i W_{ij}^1 v_i + \sum_k W_{jk}^2 h_k^2\right)$$

$$P(h_k^2 = 1|\mathbf{h}^1, \mathbf{h}^3) = \sigma\left(\sum_j W_{jk}^2 h_j^1 + \sum_m W_{km}^3 h_m^3\right)$$

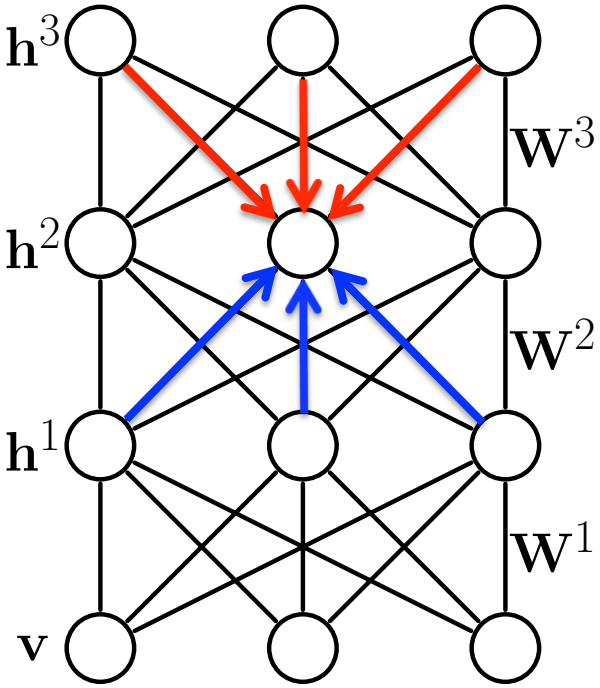
$$P(h_m^3 = 1|\mathbf{h}^2) = \sigma\left(\sum_k W_{km}^3 h_k^2\right)$$

- Note that exact computation of $P(\mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3|\mathbf{v})$ is intractable.

Mathematical Formulation

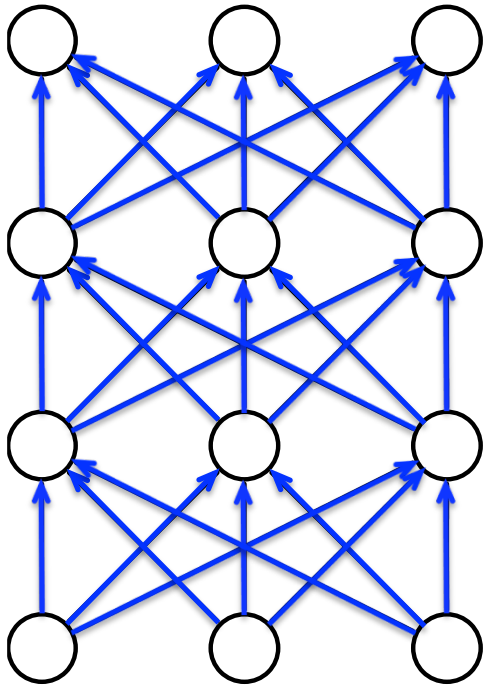
$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \mathbf{h}^{1\top} W^2 \mathbf{h}^2 + \mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$

Deep Boltzmann Machine

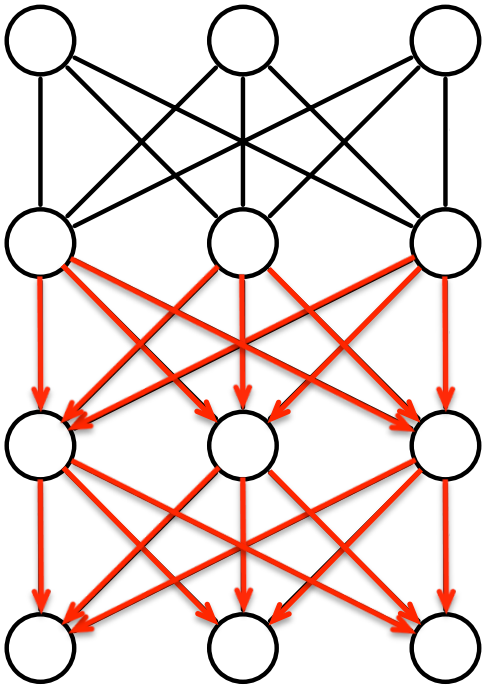


Input

Neural Network Output



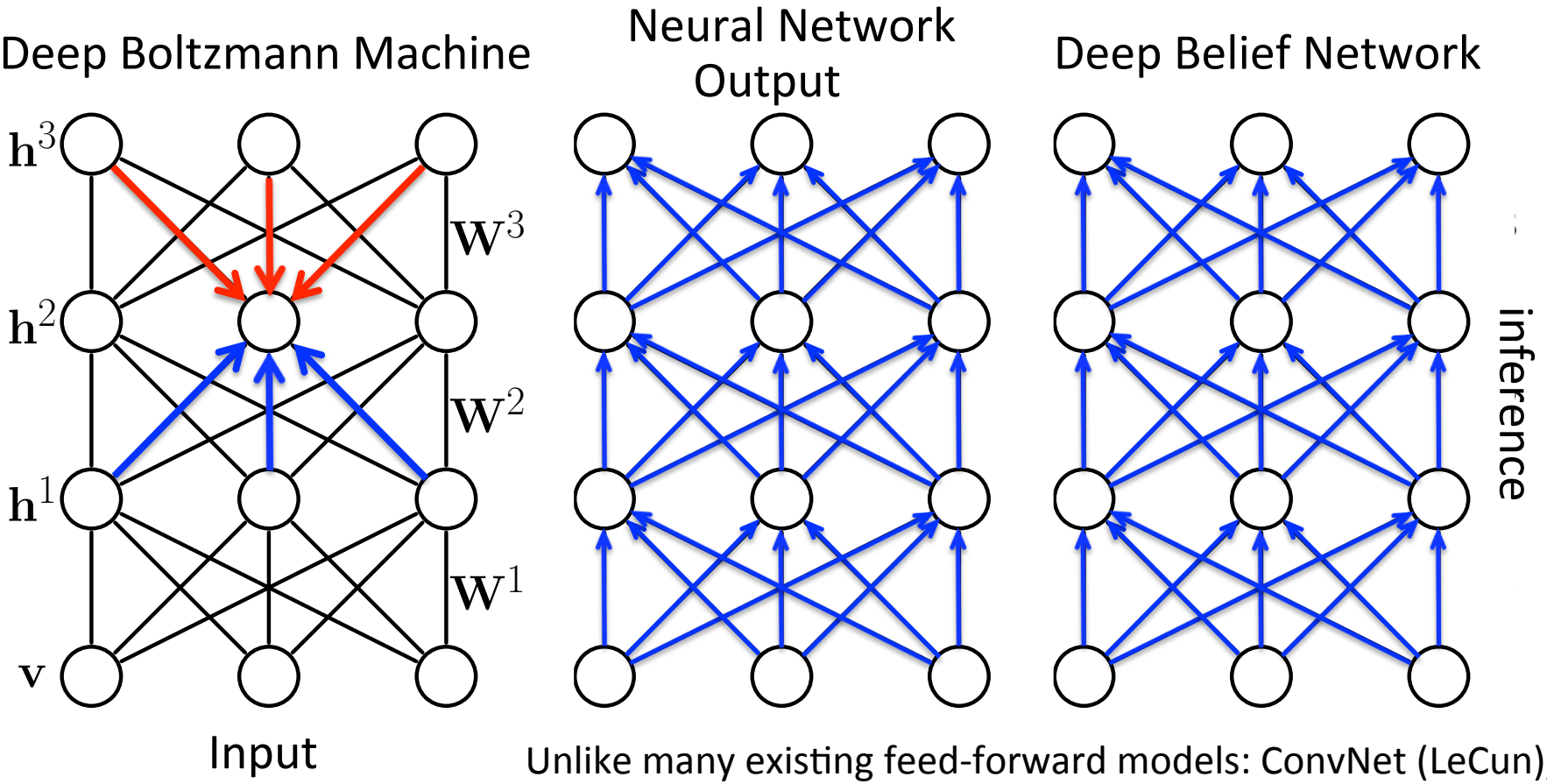
Deep Belief Network



Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio), Deep Belief Nets (Hinton)

Mathematical Formulation

$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + {\mathbf{h}^1}^\top W^2 \mathbf{h}^2 + {\mathbf{h}^2}^\top W^3 \mathbf{h}^3\right]$$

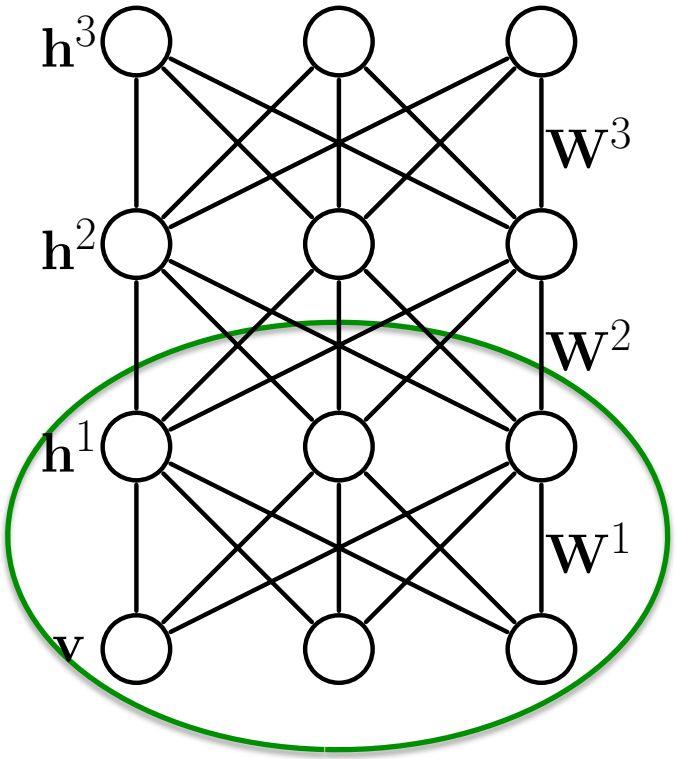


Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio), Deep Belief Nets (Hinton)

Mathematical Formulation

$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \mathbf{h}^{1\top} W^2 \mathbf{h}^2 + \mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$

Deep Boltzmann Machine



$\theta = \{W^1, W^2, W^3\}$ model parameters

- Dependencies between hidden variables.

Maximum likelihood learning:

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathrm{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{1\top}] - \mathrm{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{1\top}]$$

Problem: Both expectations are intractable!

Learning rule for undirected graphical models: MRFs, CRFs, Factor graphs.

Previous Work

Many approaches for learning Boltzmann machines have been proposed over the last 20 years:

- Hinton and Sejnowski (1983),
- Peterson and Anderson (1987)
- Galland (1991)
- Kappen and Rodriguez (1998)
- Lawrence, Bishop, and Jordan (1998)
- Tanaka (1998)
- Welling and Hinton (2002)
- Zhu and Liu (2002)
- Welling and Teh (2003)
- Yasuda and Tanaka (2009)

Real-world applications – thousands of hidden and observed variables with millions of parameters.

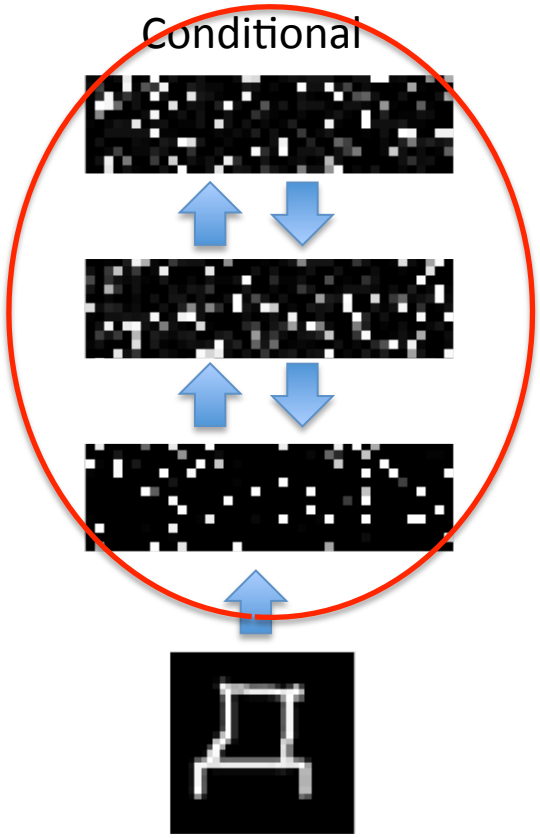
Many of the previous approaches were not successful for learning general Boltzmann machines with **hidden variables.**

Algorithms based on Contrastive Divergence, Score Matching, Pseudo-Likelihood, Composite Likelihood, MCMC-MLE, Piecewise Learning, cannot handle multiple layers of hidden variables.

New Learning Algorithm

Posterior Inference

Conditional



Approximate conditional

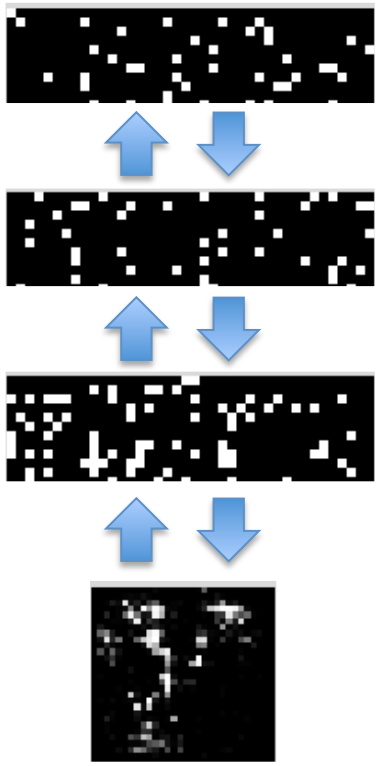
$P_{data}(\mathbf{h}|\mathbf{v})$

Simulate from the Model

Approximate the joint distribution

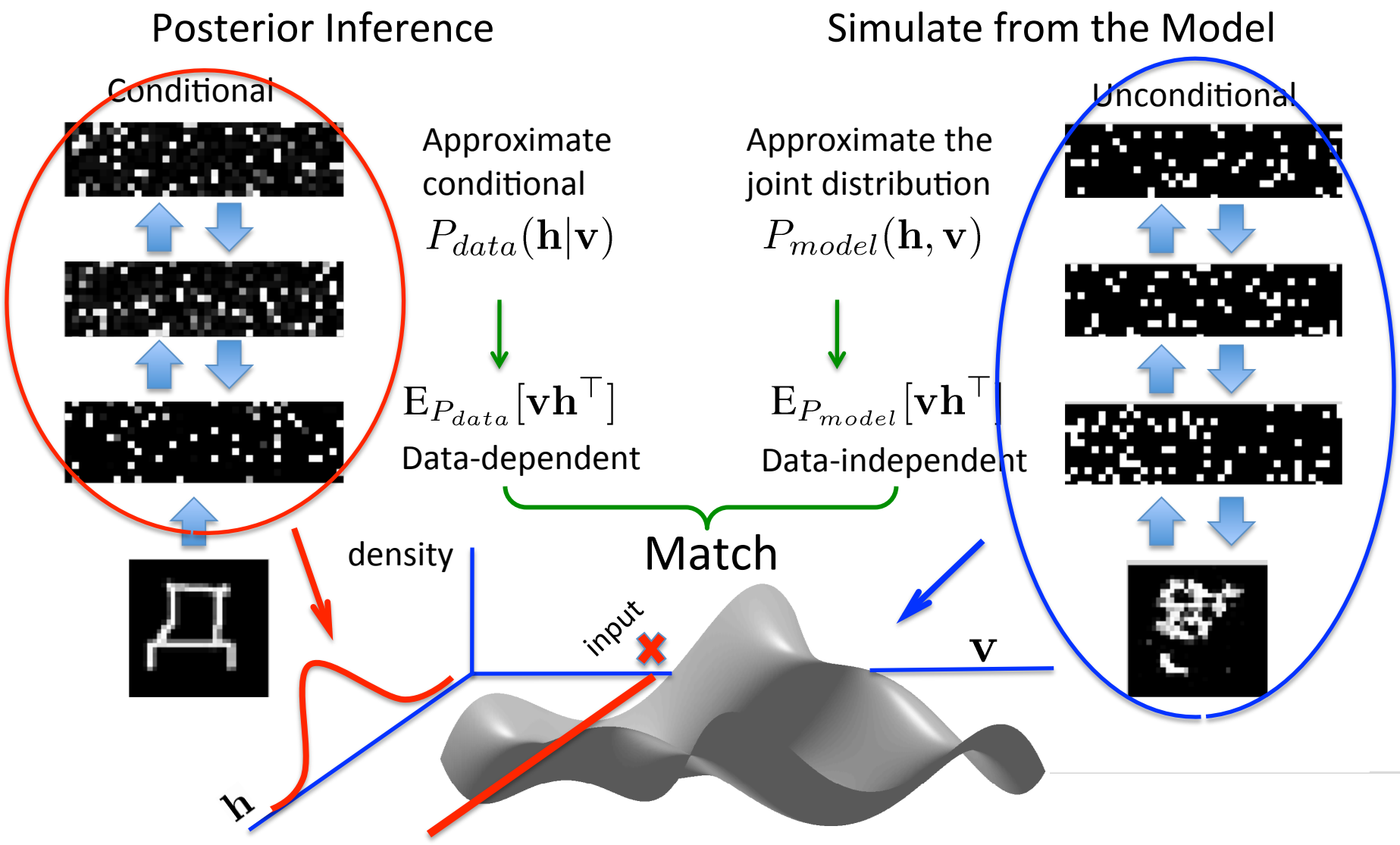
$P_{model}(\mathbf{h}, \mathbf{v})$

Unconditional



(Salakhutdinov, 2008; NIPS 2009)

New Learning Algorithm

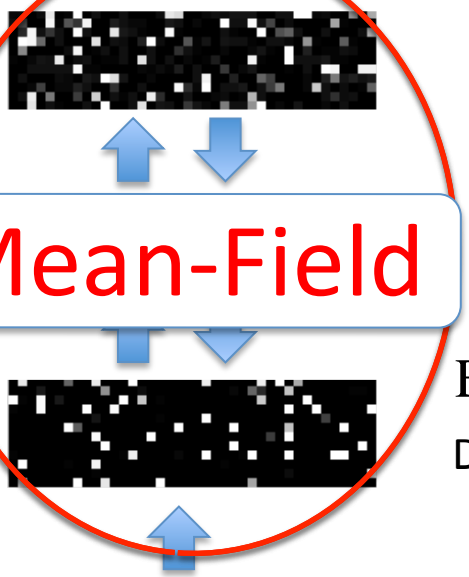


New Learning Algorithm

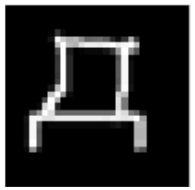
Posterior Inference

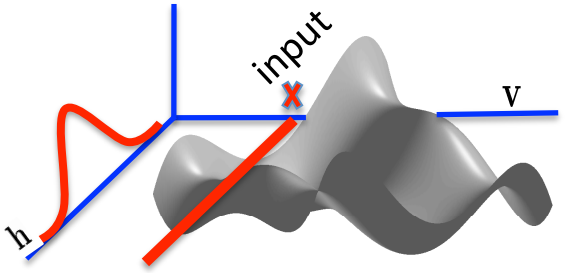
Simulate from the Model

Conditional



Mean-Field





$\mathrm{E}_{P_{data}}[\mathbf{v}\mathbf{h}^\top]$

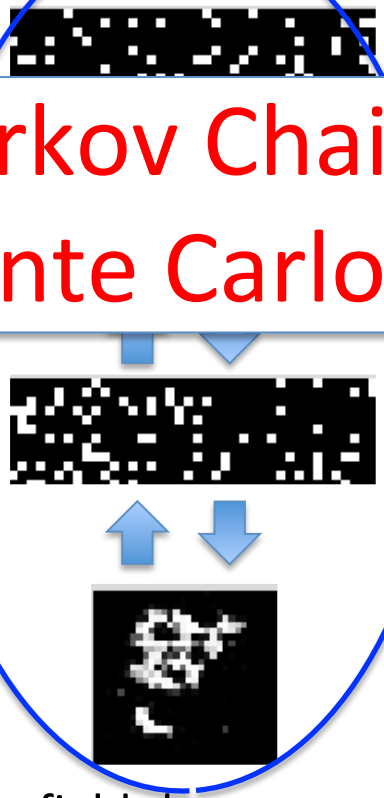
Data-dependent

$\mathrm{E}_{P_{model}}[\mathbf{v}\mathbf{h}^\top]$

Data-independent

Match

Unconditional



Markov Chain Monte Carlo

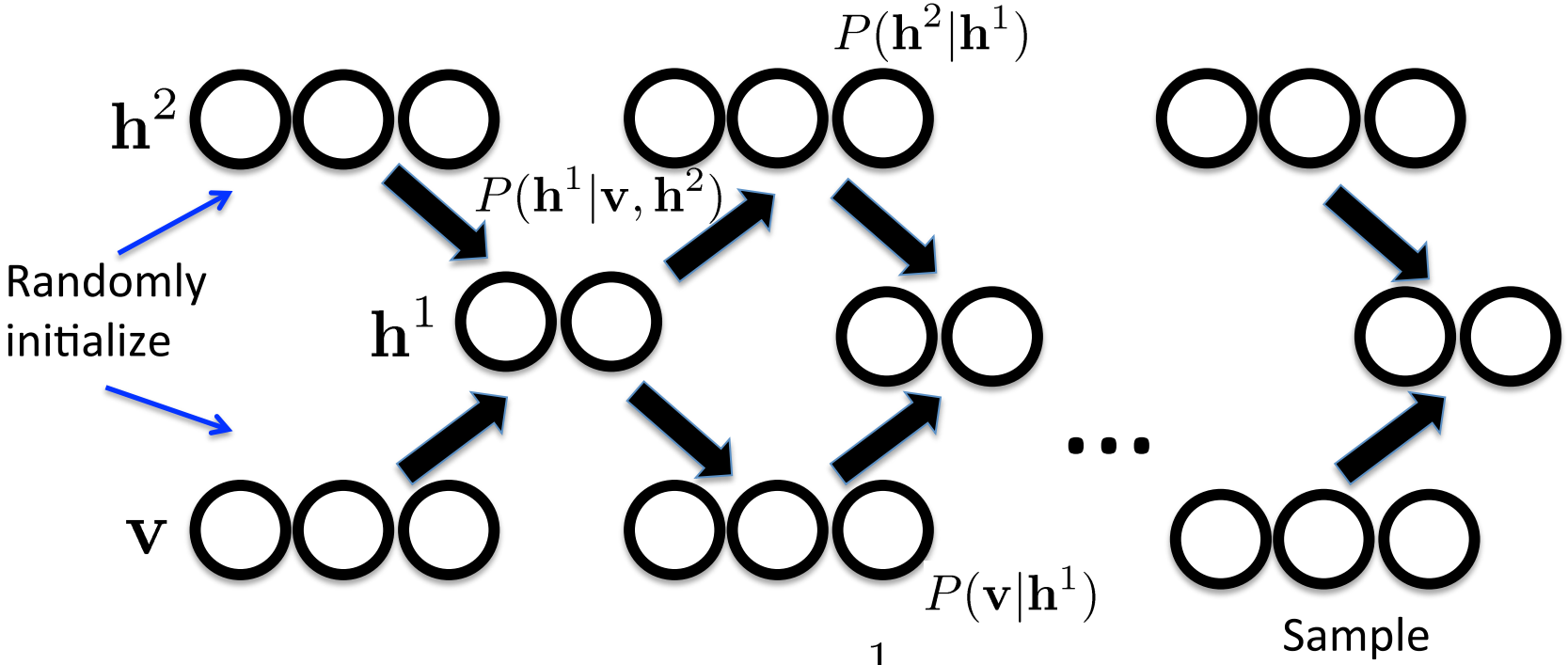
Key Idea of Our Approach:

Data-dependent: **Variational Inference**, mean-field theory

Data-independent: **Stochastic Approximation**, MCMC based

Sampling from DBMs

Sampling from two-hidden layer DBM by running a Markov chain:

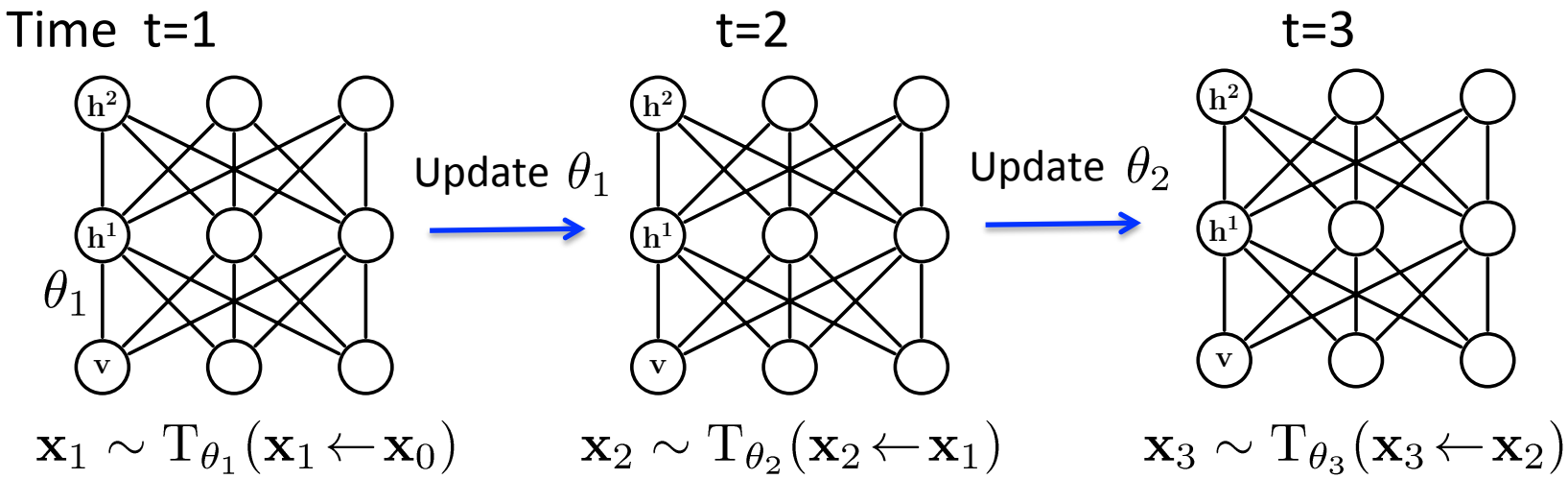


$$P(h_m^1 = 1|\mathbf{v},\mathbf{h}^2) = \frac{1}{1+\exp(-\sum_i W_{im}^1 v_i - \sum_j W_{mj}^2 h_j^2)}$$

$$P(h_j^2 = 1|\mathbf{h}^1) = \frac{1}{1+\exp(-\sum_m W_{mj}^2 h_m^1)}$$

$$P(v_i = 1|\mathbf{h}^1) = \frac{1}{1+\exp(-\sum_m W_{im}^1 h_m^1)}$$

Stochastic Approximation



Update θ_t and $\mathbf{x}_t$ sequentially, where $\mathbf{x} = \{\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2\}$

- Generate $\mathbf{x}_t \sim \mathrm{T}_{\theta_t}(\mathbf{x}_t \leftarrow \mathbf{x}_{t-1})$ by simulating from a Markov chain that leaves P_{θ_t} invariant (e.g. Gibbs or M-H sampler)
- Update θ_t by replacing intractable $\mathrm{E}_{P_{\theta_t}}[\mathbf{v}\mathbf{h}^\top]$ with a point estimate $[\mathbf{v}_t\mathbf{h}_t^\top]$

In practice we simulate several Markov chains in parallel.

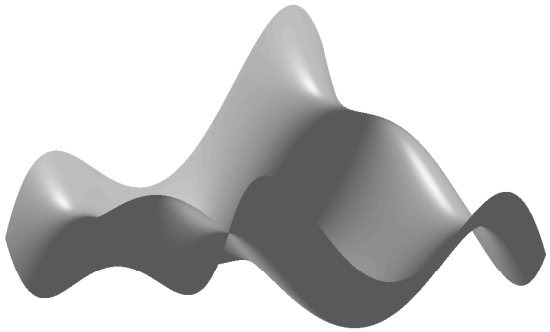
Robbins and Monro, Ann. Math. Stats, 1957
L. Younes, Probability Theory 1989

Stochastic Approximation

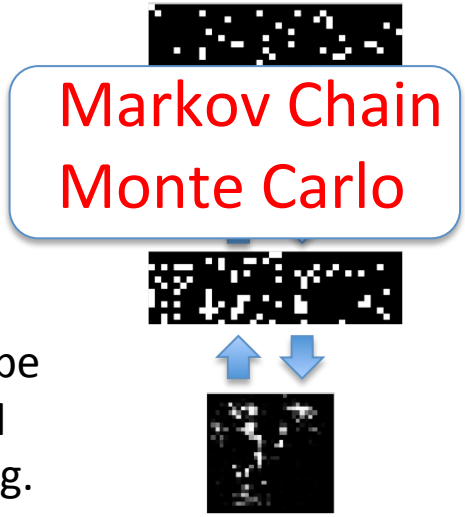
Update rule decomposes:

$$\theta_{t+1} = \theta_t + \alpha_t \underbrace{\left(\mathrm{E}_{P_{data}}[\mathbf{v}\mathbf{h}^\top] - \mathrm{E}_{P_{\theta_t}}[\mathbf{v}\mathbf{h}^\top] \right)}_{\text{True gradient}} + \alpha_t \underbrace{\left(\mathrm{E}_{P_{\theta_t}}[\mathbf{v}\mathbf{h}^\top] - \frac{1}{M} \sum_{m=1}^{M} \mathbf{v}_t^{(m)} \mathbf{h}_t^{(m)\top} \right)}_{\text{Noise term } \epsilon_t}$$

Almost sure convergence guarantees as learning rate $\alpha_t \to 0$



Problem: High-dimensional data: the energy landscape is highly multimodal



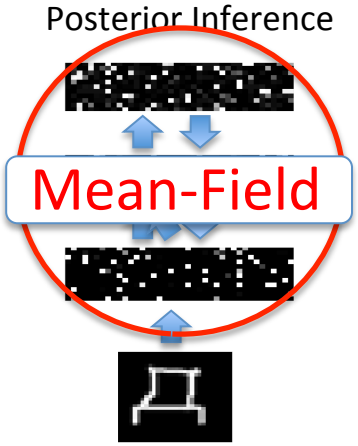
Key insight: The transition operator can be any valid transition operator – Tempered Transitions, Parallel/Simulated Tempering.

Connections to the theory of stochastic approximation and adaptive MCMC.

Variational Inference

Approximate intractable distribution $P_\theta(\mathbf{h}|\mathbf{v})$ with simpler, tractable distribution $Q_\mu(\mathbf{h}|\mathbf{v})$:

$$\log P_\theta(\mathbf{v}) = \log \sum_{\mathbf{h}} P_\theta(\mathbf{h}, \mathbf{v}) = \log \sum_{\mathbf{h}} Q_\mu(\mathbf{h}|\mathbf{v}) \frac{P_\theta(\mathbf{h}, \mathbf{v})}{Q_\mu(\mathbf{h}|\mathbf{v})}$$



$$\geq \sum_{\mathbf{h}} Q_\mu(\mathbf{h}|\mathbf{v}) \log \frac{P_\theta(\mathbf{h}, \mathbf{v})}{Q_\mu(\mathbf{h}|\mathbf{v})}$$

$$= \sum_{\mathbf{h}} Q_\mu(\mathbf{h}|\mathbf{v}) \underbrace{\log P^*_\theta(\mathbf{h}, \mathbf{v})}_{\mathbf{v}^\top W^1 \mathbf{h}^1 + {\mathbf{h}^1}^\top W^2 \mathbf{h}^2 + {\mathbf{h}^2}^\top W^3 \mathbf{h}^3} - \log \mathcal{Z}(\theta) + \sum_{\mathbf{h}} Q_\mu(\mathbf{h}|\mathbf{v}) \log \frac{1}{Q_\mu(\mathbf{h}|\mathbf{v})}$$

Variational Lower Bound

$$= \log P_\theta(\mathbf{v}) - \mathrm{KL}\big(Q_\mu(\mathbf{h}|\mathbf{v}) || P_\theta(\mathbf{h}|\mathbf{v})\big)$$

$$\mathrm{KL}(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$$

Minimize KL between approximating and true distributions with respect to variational parameters μ .

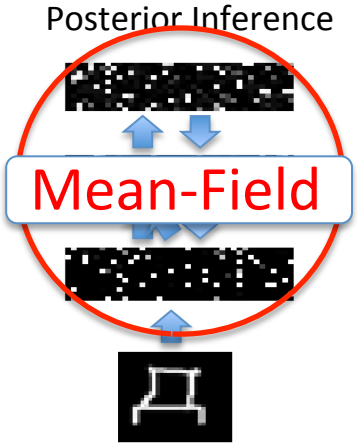
(Salakhutdinov, 2008; Salakhutdinov & Larochelle, AI & Statistics 2010)

Variational Inference

Approximate intractable distribution $P_\theta(\mathbf{h}|\mathbf{v})$ with simpler, tractable distribution $Q_\mu(\mathbf{h}|\mathbf{v})$:

$$\mathrm{KL}(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$$

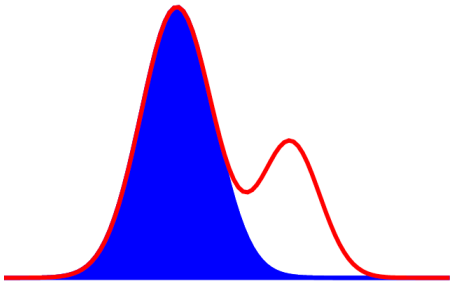
$$\log P_\theta(\mathbf{v}) \geq \underbrace{\log P_\theta(\mathbf{v}) - \mathrm{KL}\big(Q_\mu(\mathbf{h}|\mathbf{v})||P_\theta(\mathbf{h}|\mathbf{v})\big)}_{\text{Variational Lower Bound}}$$



Mean-Field: Choose a fully factorized distribution:

$$Q_\mu(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{F} q(h_j|\mathbf{v}) \text{ with } q(h_j = 1|\mathbf{v}) = \mu_j$$

Variational Inference: Maximize the lower bound w.r.t. Variational parameters μ.



Nonlinear fixed-point equations:

$$\mu_j^{(1)} = \sigma\Big(\sum_i W_{ij}^1 v_i + \sum_k W_{jk}^2 \mu_k^{(2)}\Big)$$
$$\mu_k^{(2)} = \sigma\Big(\sum_j W_{jk}^2 \mu_j^{(1)} + \sum_m W_{km}^3 \mu_m^{(3)}\Big)$$
$$\mu_m^{(3)} = \sigma\Big(\sum_k W_{km}^3 \mu_k^{(2)}\Big)$$

Variational Inference

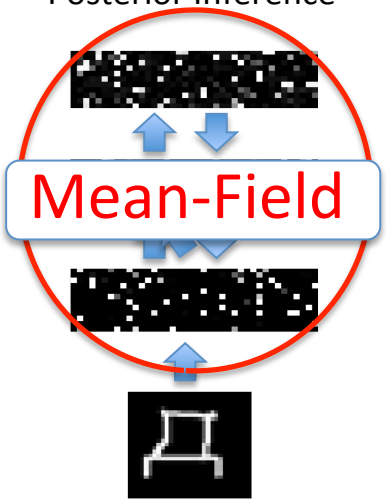
Approximate intractable distribution $P_\theta(\mathbf{h}|\mathbf{v})$ with simpler, tractable distribution $Q_\mu(\mathbf{h}|\mathbf{v})$:

$$\mathrm{KL}(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$$

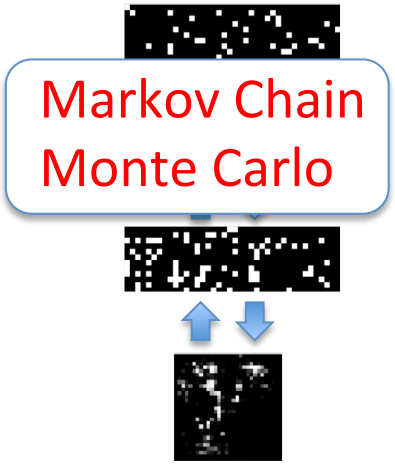
$$\log P_\theta(\mathbf{v}) \geq \log P_\theta(\mathbf{v}) - \mathrm{KL}\big(Q_\mu(\mathbf{h}|\mathbf{v})||P_\theta(\mathbf{h}|\mathbf{v})\big)$$

Variational Lower Bound

Posterior Inference



Unconditional Simulation



1. Variational Inference: Maximize the lower bound w.r.t. variational parameters

2. MCMC: Apply stochastic approximation to update model parameters

Almost sure convergence guarantees to an asymptotically stable point.

Variational Inference

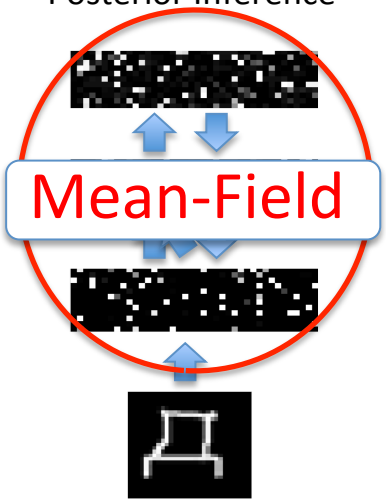
Approximate intractable distribution $P_\theta(\mathbf{h}|\mathbf{v})$ with simpler, tractable distribution $Q_\mu(\mathbf{h}|\mathbf{v})$:

$$\mathrm{KL}(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$$

$$\log P_\theta(\mathbf{v}) \geq \log P_\theta(\mathbf{v}) - \mathrm{KL}\big(Q_\mu(\mathbf{h}|\mathbf{v})||P_\theta(\mathbf{h}|\mathbf{v})\big)$$

Variational Lower Bound

Posterior Inference



Unconditional Simulation

1. V wer bou

Fast Inference

2. M to u

Learning can scale to millions of examples

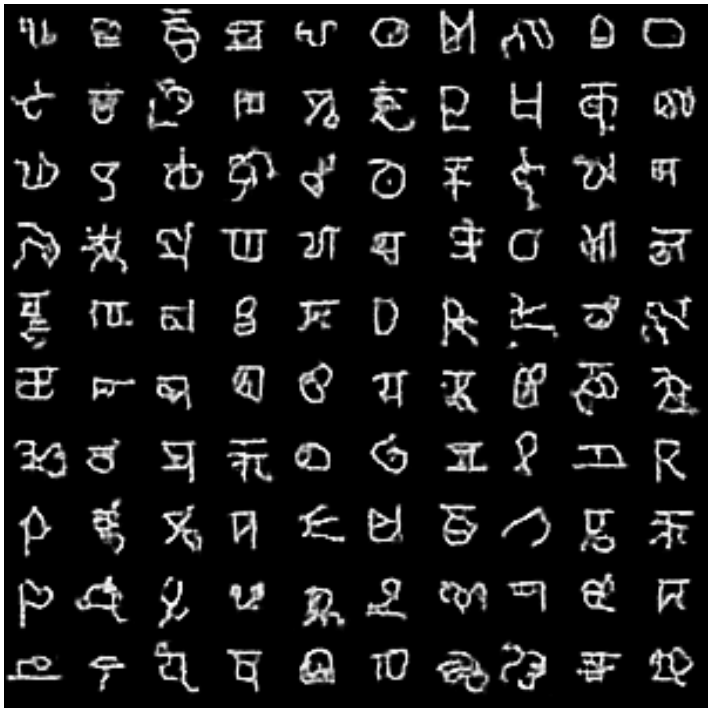
Almost sure convergence guarantees to an asymptotically stable point.

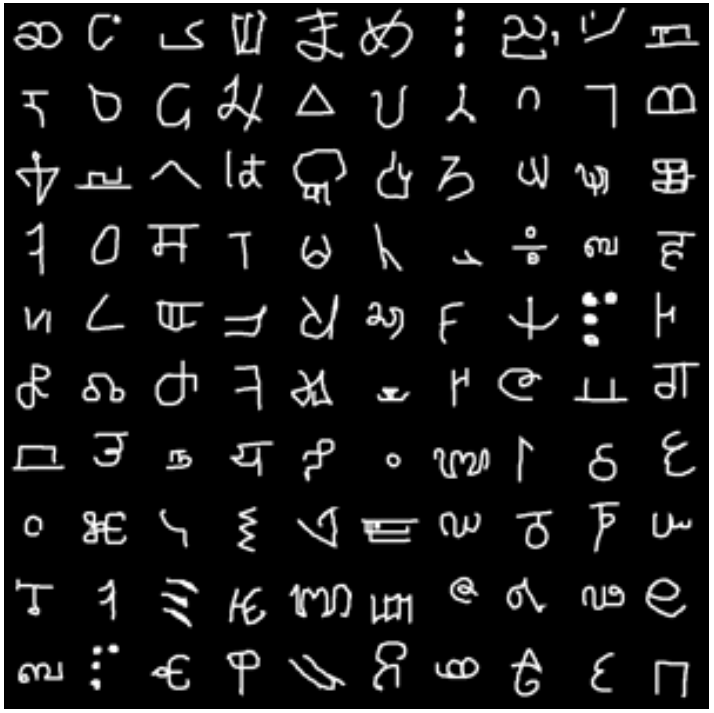
Good Generative Model?

Handwritten Characters

Good Generative Model?

Handwritten Characters





Good Generative Model?

Handwritten Characters

Simulated

Real Data

Good Generative Model?

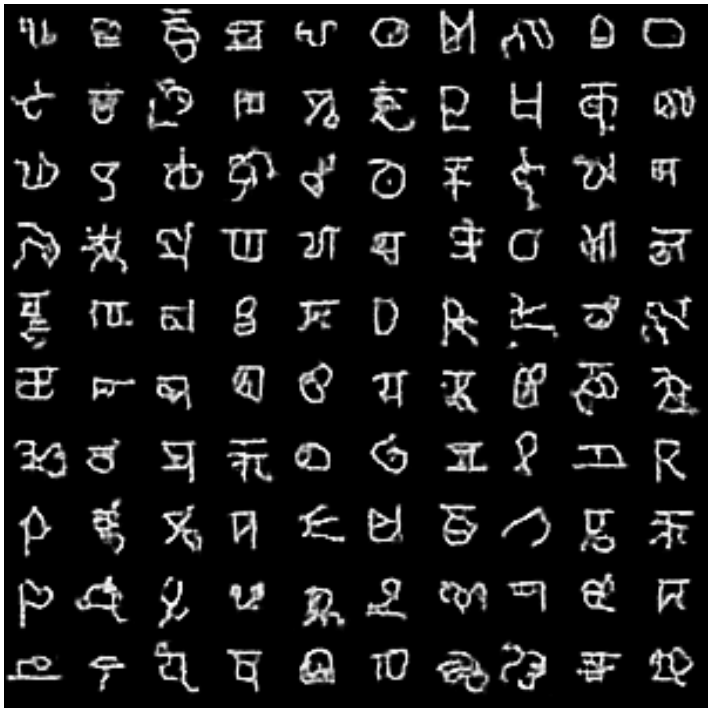
Handwritten Characters

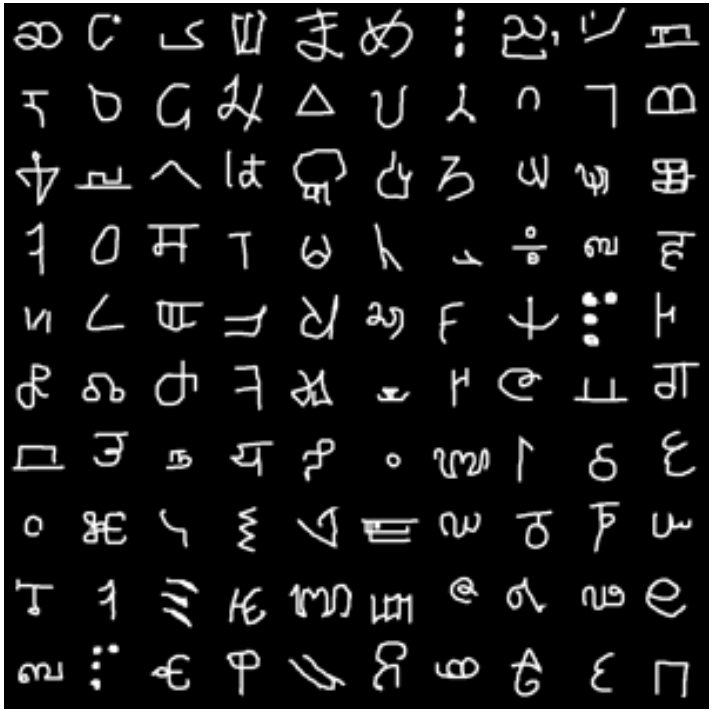
Real Data

Simulated

Good Generative Model?

Handwritten Characters





Good Generative Model?

MNIST Handwritten Digit Dataset

Handwriting Recognition

MNIST Dataset
60,000 examples of 10 digits

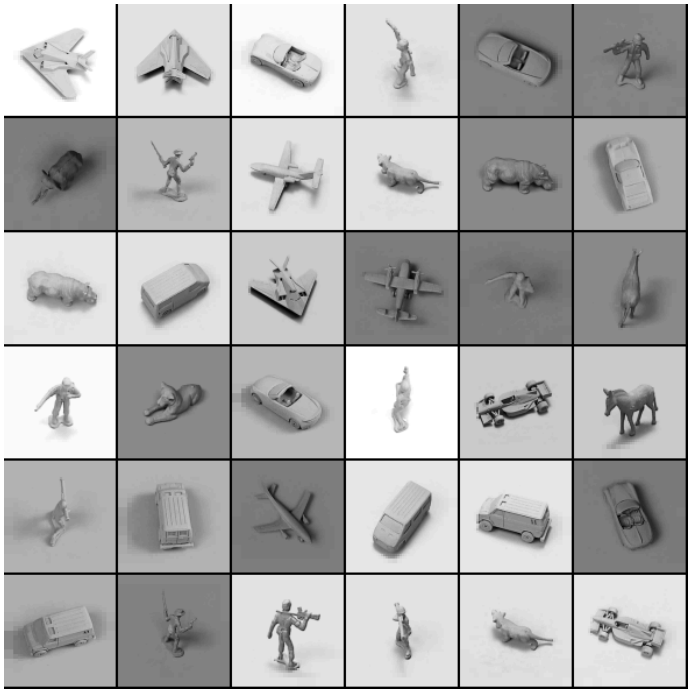
Learning Algorithm	Error
Logistic regression	12.0%
K-NN	3.09%
Neural Net (Platt 2005)	1.53%
SVM (Decoste et.al. 2002)	1.40%
Deep Autoencoder (Bengio et. al. 2007)	1.40%
Deep Belief Net (Hinton et. al. 2006)	1.20%
DBM	**0.95%**

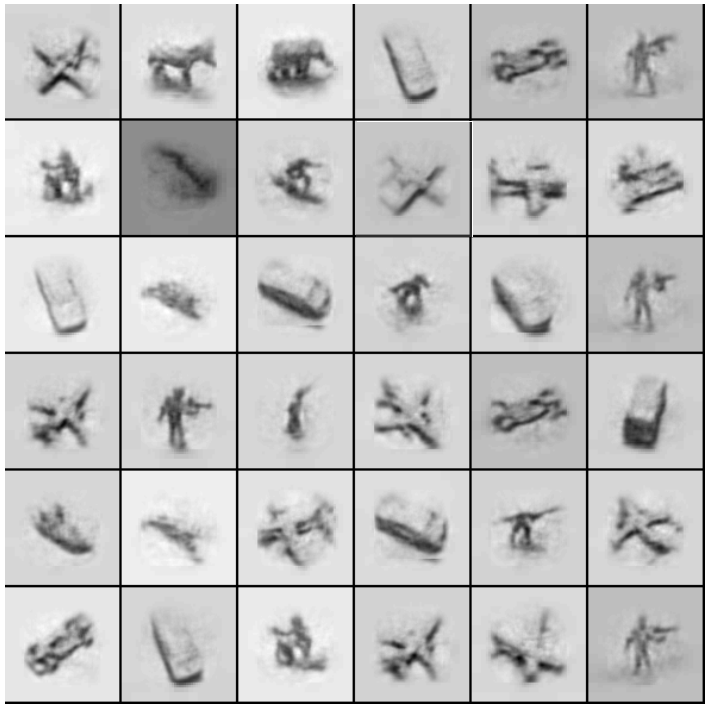
Optical Character Recognition
42,152 examples of 26 English letters

Learning Algorithm	Error
Logistic regression	22.14%
K-NN	18.92%
Neural Net	14.62%
SVM (Larochelle et.al. 2009)	9.70%
Deep Autoencoder (Bengio et. al. 2007)	10.05%
Deep Belief Net (Larochelle et. al. 2009)	9.68%
DBM	**8.40%**

Permutation-invariant version.

Generative Model of 3-D Objects



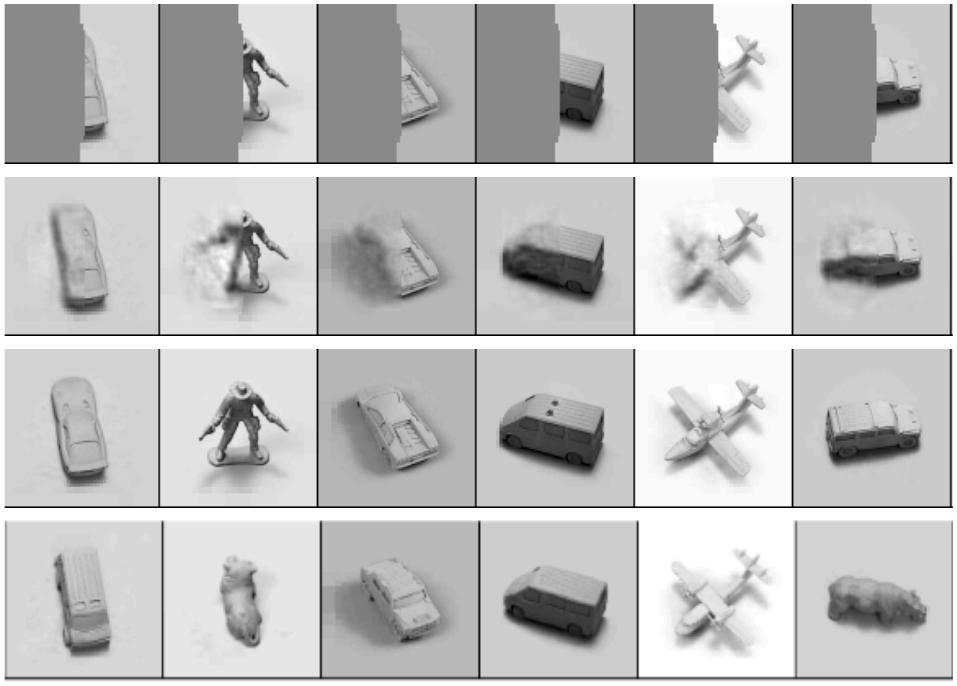


24,000 examples, 5 object categories, 5 different objects within each category, 6 lightning conditions, 9 elevations, 18 azimuths.

3-D Object Recognition

Learning Algorithm	Error
Logistic regression	22.5%
K-NN (LeCun 2004)	18.92%
SVM (Bengio & LeCun 2007)	11.6%
Deep Belief Net (Nair & Hinton 2009)	9.0%
DBM	**7.2%**

Pattern Completion

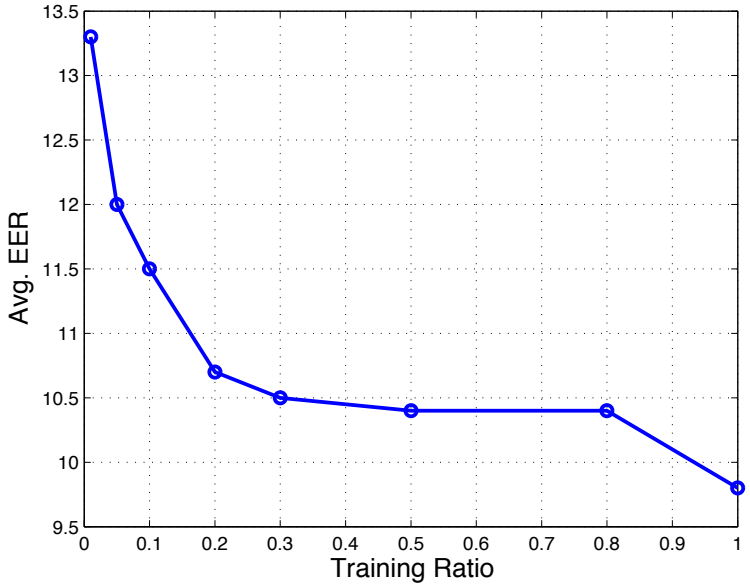


Permutation-invariant version.

Spoken Query Detection

- 630 speaker TIMIT corpus: 3,696 training and 944 test utterances.
- 10 query keywords were randomly selected and 10 examples of each keyword were extracted from the training set.
- **Goal**: For each keyword, rank all 944 utterances based on the utterance's probability of containing that keyword.
- Performance measure: The average equal error rate (EER).

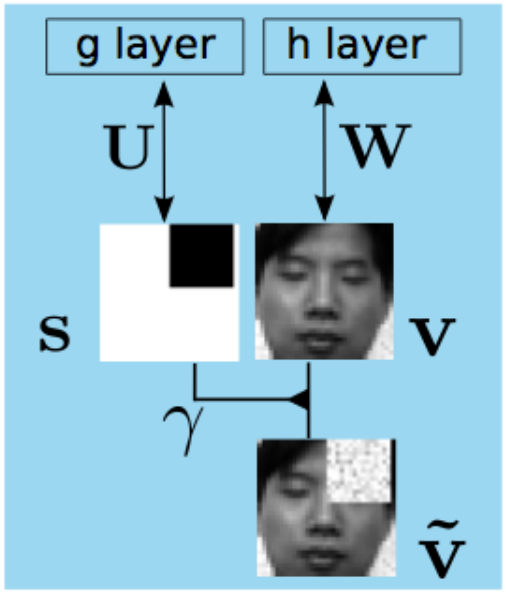
Learning Algorithm	AVG EER
GMM Unsupervised	16.4%
DBM Unsupervised	14.7%
DBM (1% labels)	13.3%
DBM (30% labels)	10.5%
DBM (100% labels)	9.7%



(Yaodong Zhang et.al. ICASSP 2012)

Robust Boltzmann Machines

• Build more complex models that can deal with occlusions or structured noise.



Inferred

Observed

Gaussian RBM, modeling clean faces

Binary RBM modeling occlusions

$$E = \frac{1}{2}\sum_i \frac{(v_i - b_i)^2}{\sigma_i^2} - \mathbf{v}^\top W\mathbf{h} - \mathbf{s}^\top U\mathbf{g}$$
$$+\frac{1}{2}\sum_i \gamma_i s_i (v_i - \tilde{v}_i)^2 + \frac{1}{2}\sum_i \frac{(\tilde{v}_i - \tilde{b}_i)^2}{\tilde{\sigma}_i^2}$$

Binary pixel-wise Mask

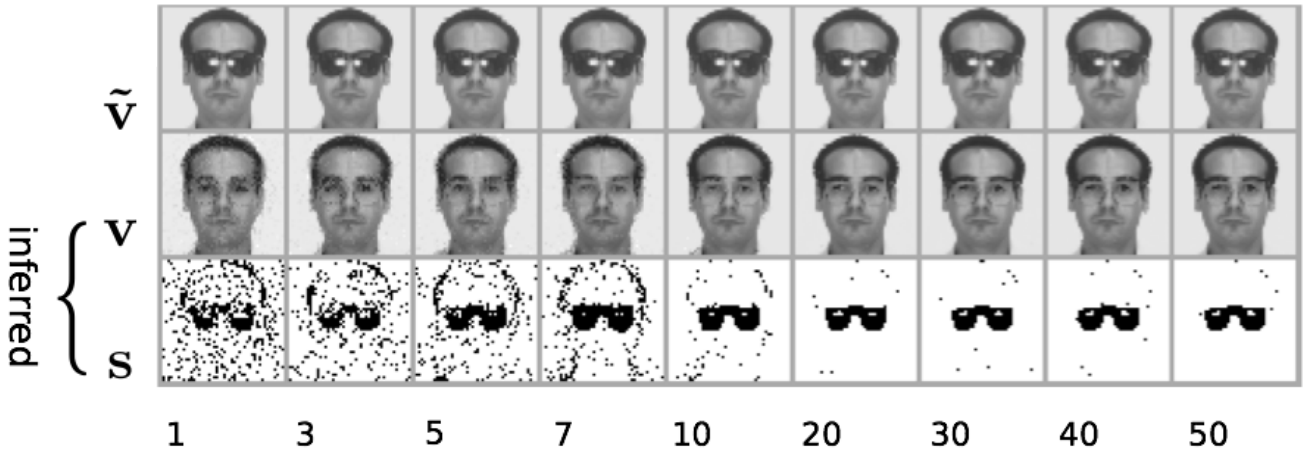
Gaussian noise

Relates to Le Roux, Heess, Shotton, and Winn, Neural Computation, 2011

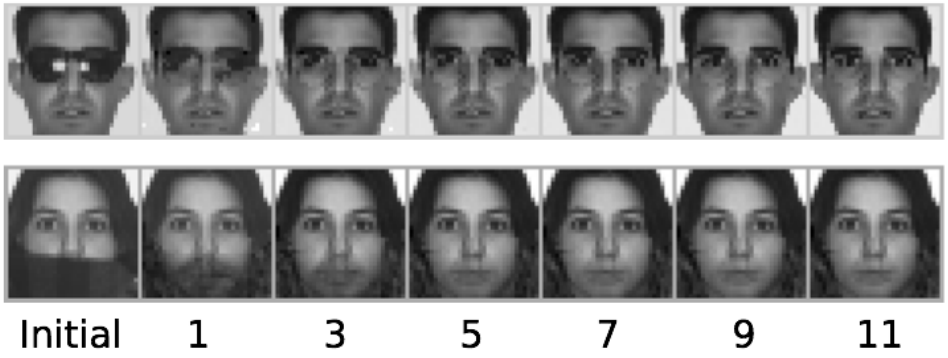
Eslami, Heess, Winn, CVPR 2012

Tang, Salakhutdinov, and Hinton, CVPR 2012

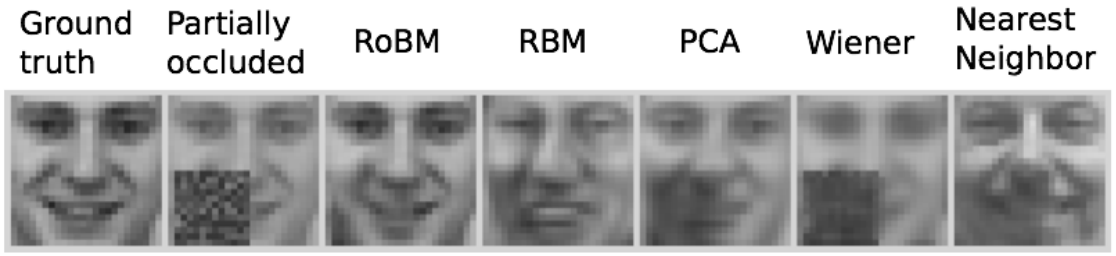
Robust Boltzmann Machines



Internal States of RoBM during learning.



Inference on the test subjects

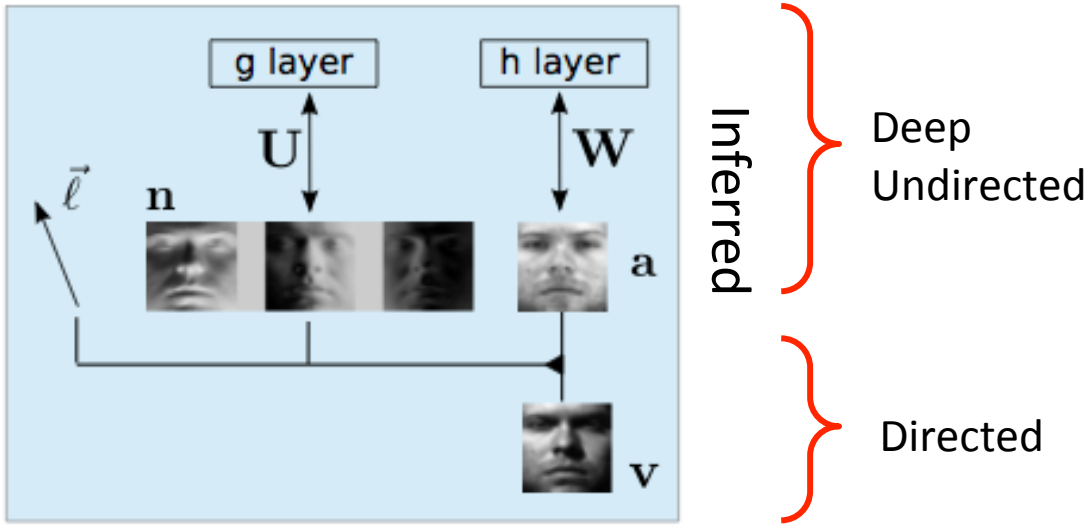


Comparing to Other Denoising Algorithms

Deep Lambertian Network

Consider More Complex Models: undirected + directed models.

Deep Lambertian Net



Combines the elegant properties of the Lambertian model with the Gaussian RBMs (and Deep Belief Nets, Deep Boltzmann Machines).

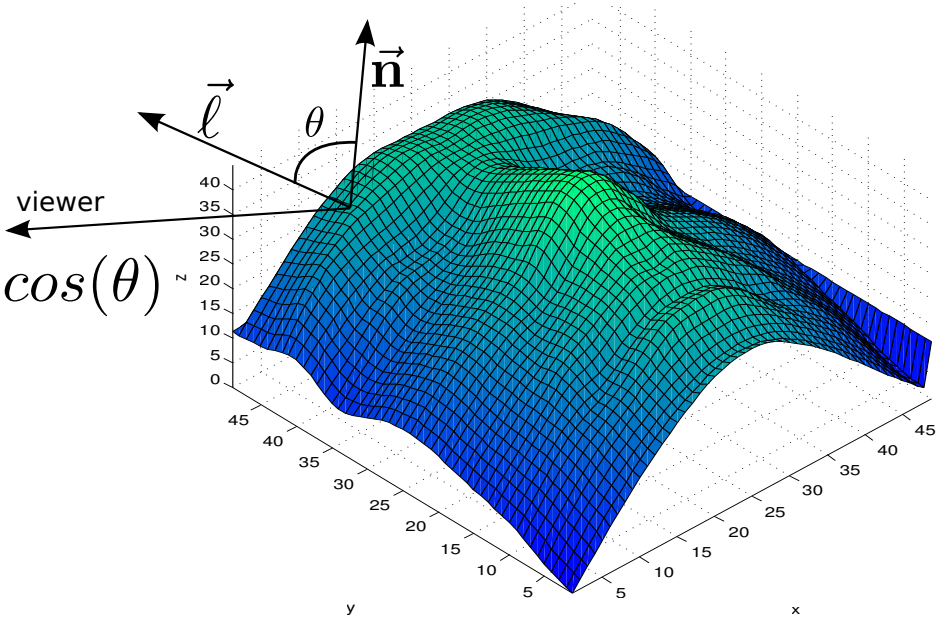
Tang, Salakhutdinov, and Hinton, ICML 2012

Lambertian Reflectance Model

- A simple model of the image formation process.
- Albedo is the diffuse reflectivity of a surface, material dependent, illumination independent
- Images with different illumination can be generated by varying light directions

Lambertian Reflectance Model

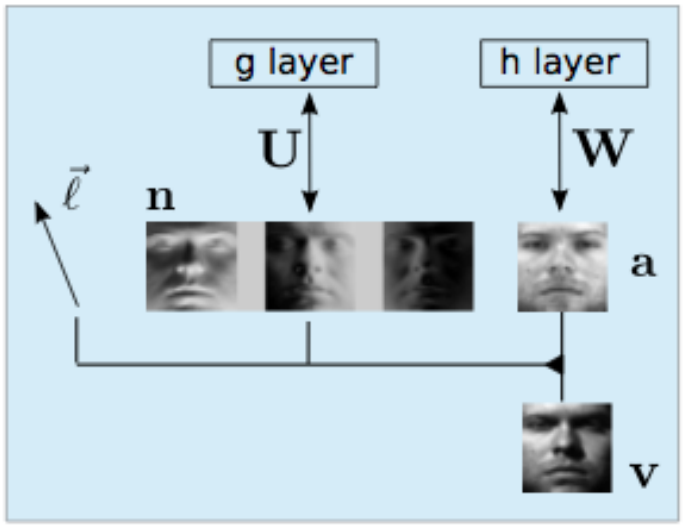
$$I_{viewer} = a \times |\vec{\ell}||\vec{\mathbf{n}}|cos(\theta)$$



Deep Lambertian Networks

Model Details:

Deep Lambertian Net



Inferred

Observed

Image albedo

Surface Normals

Light source

$$P(\mathbf{v}|\mathbf{a},\mathbf{N},l) = \prod_{i \in \text{pixels}} \mathcal{N}\left(v_i | a_i(\mathbf{n}_i^\top l)\right)$$

$$\mathbf{a} \in \mathbb{R}^N, \quad \mathbf{N} \in \mathbb{R}^{N\times 3}, \quad l \in \mathbb{R}^3$$

$$\begin{aligned} P(\mathbf{a}) &\sim GRBM(\mathbf{a}), \\ P(\mathbf{N}) &\sim GRBM(\mathbf{N}), \\ P(l) &\sim \mathcal{N}(\mu, \mathbf{\Lambda}) \end{aligned}$$

Inference: Gibbs sampler.

Learning: Stochastic Approximation

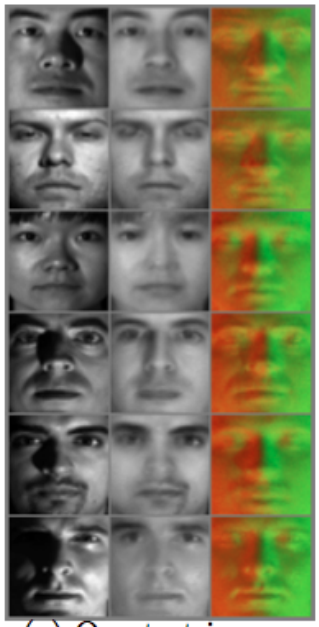
Yale B Extended Face Dataset

- 38 subjects, ~ 45 images of varying illuminations per subject, divided into 4 subsets of increasing illumination variations
- 28 subjects for training, 10 (original Yale Database) for testing
- Toronto Face Database is used to pretrain the “albedo DBN”

Deep Lambertian Networks

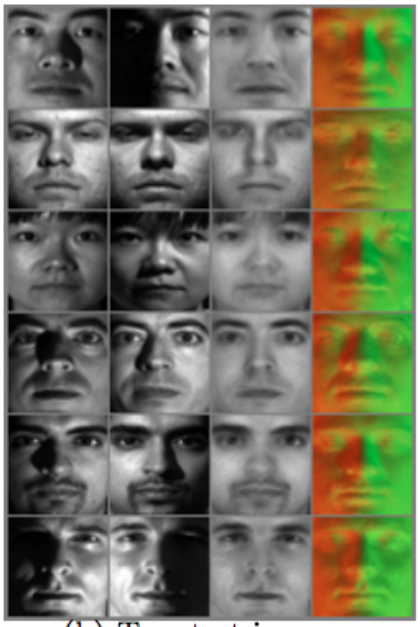
Yale B Extended Database

One Test Image



(a) One test image.

Two Test Images



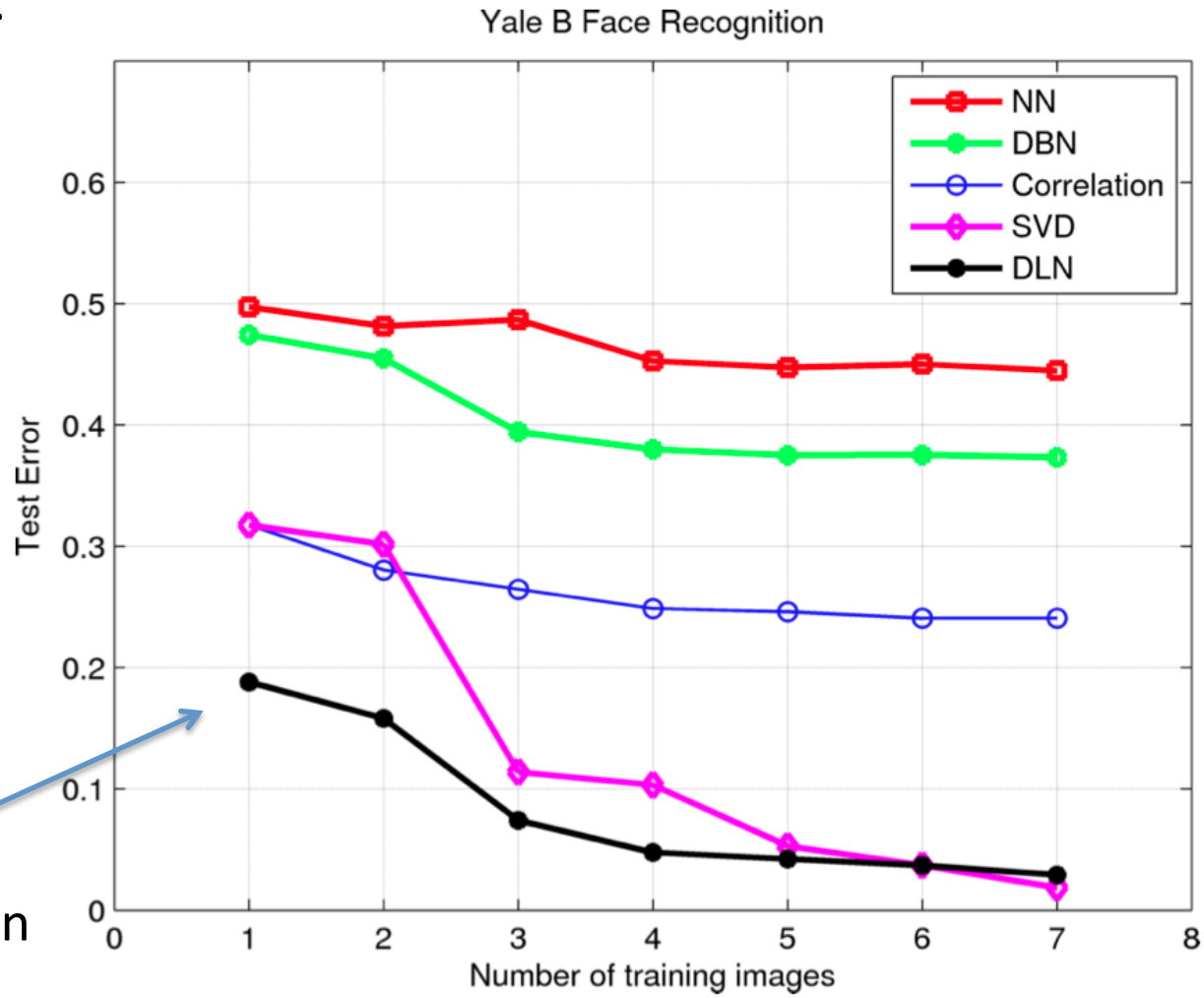
(b) Two test images.

Face Relighting

(c) Face Relighting.

Deep Lambertian Networks

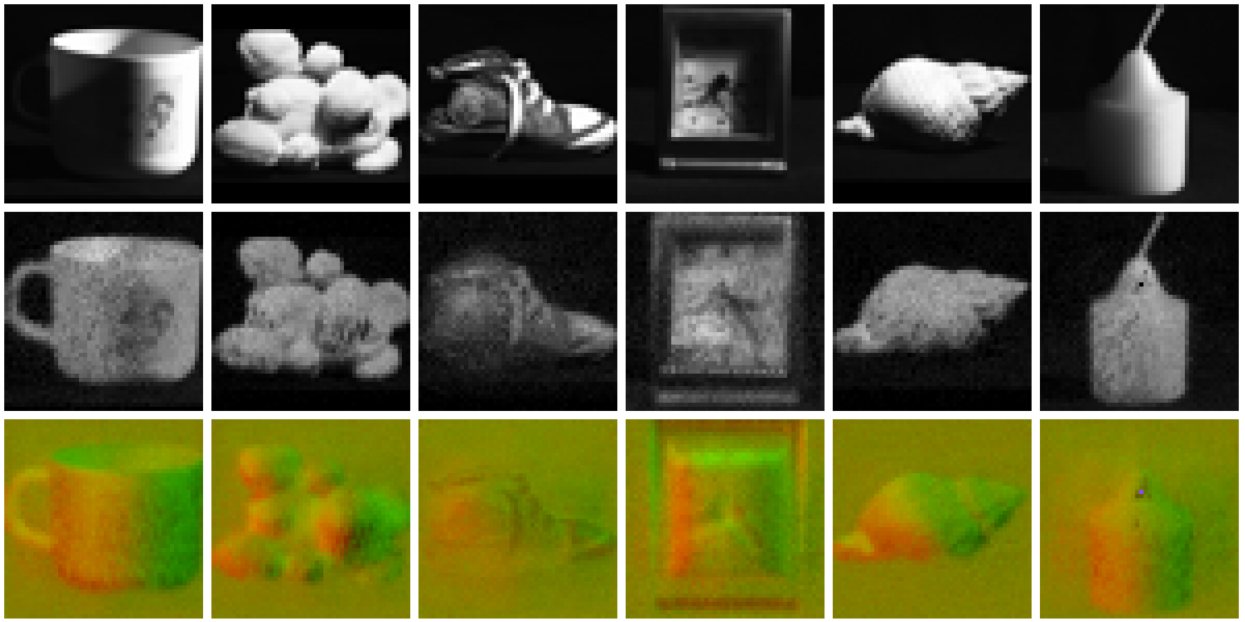
Recognition as function of the number of training images for 10 test subjects.



One-Shot Recognition

Generic Objects

- Amsterdam Library of Images (Geusebroek et al. 2005).
- For each object, 10 images taken for training, 5 for testing.



Multi-Modal Input

Learning systems that combine multiple input domains

Images

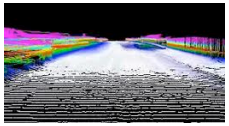
Text & Language

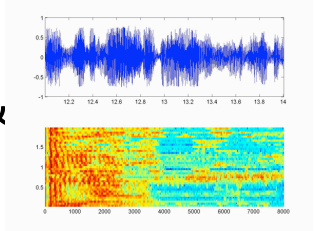
REUTERS

AP Associated Press

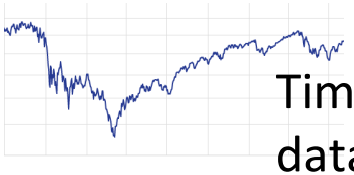
Video

Laser scans





Speech & Audio

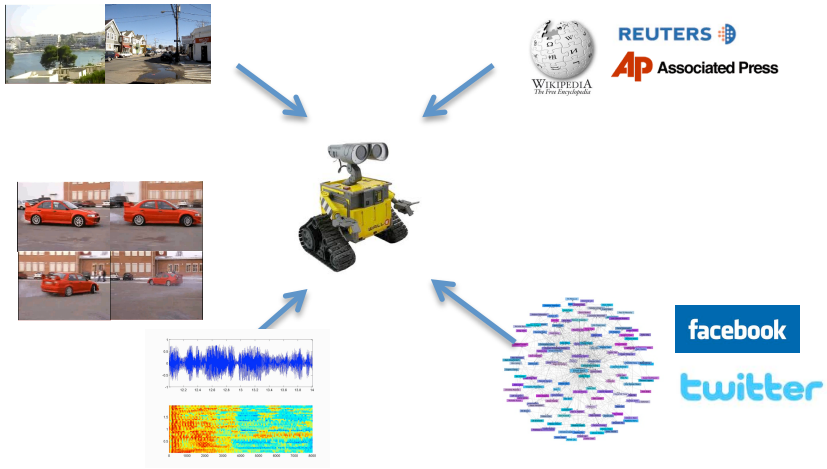


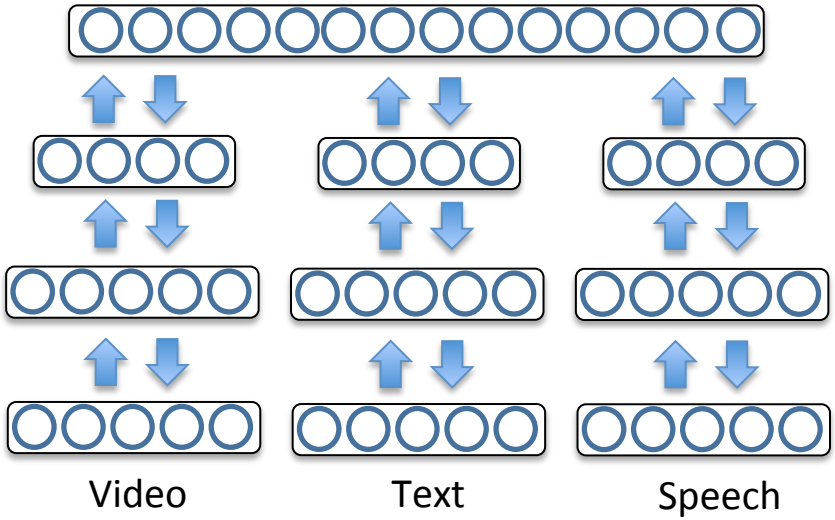
Time series data

Develop learning systems that come closer to displaying human like intelligence.

Multi-Modal Input

Learning systems that combine multiple input domains





More robust perception.

Ngiam et.al., ICML 2011 used deep autoencoders (video + speech)

- Guillaumin, Verbeek, and Schmid, CVPR 2011
- Huiskes, Thomee, and Lew, Multimedia Information Retrieval, 2010
- Xing, Yan, and Hauptmann, UAI 2005.

Training Data

pentax, k10d, kangarooisland southaustralia, sa australia australiansealion 300mm

camera, jahdakine, lightpainting, reflection doublepaneglass wowiekazowie

sandbanks, lake, lakeontario, sunset, walking, beach, purple, sky, water, clouds, overtheexcellence

top20butterflies

<no text>

mickikrimmel, mickipedia, headshot

Samples from the MIR Flickr Dataset - Creative Commons License

Multi-Modal Input

- Improve Classification

pentax, k10d, kangarooisland
southaustralia, sa australia
australiansealion 300mm

→ SEA / NOT SEA

- Fill in Missing Modalities

→ beach, sea, surf,
strand, shore,
wave, seascape,
sand, ocean, waves

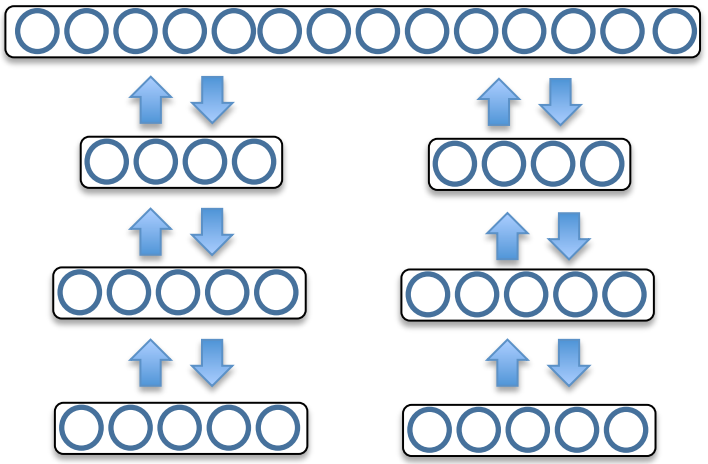
- Retrieve data from one modality when queried using data from another modality

beach, sea, surf,
strand, shore,
wave, seascape,
sand, ocean, waves

→

Srivastava and Salakhutdinov, 2012

Multi-Modal Deep Boltzmann Machine



Gaussian RBM

Dense Image features

Image

Text

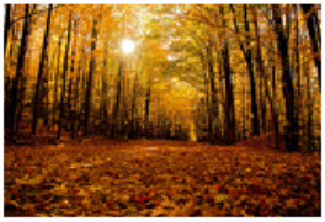
Replicated Softmax

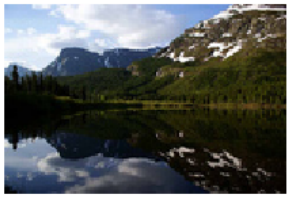
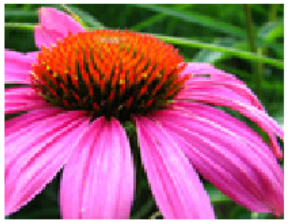
Sparse word counts

Srivastava and Salakhutdinov, 2012

Multi-Modal DBM

- Flickr Data - 1 Million images along with text tags, 25K annotated

Image	Given Tags	Generated Tags
	pentax, k10d, kangarooisland, southaustralia, sa, australia, australiansealion, 300mm	beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves
	<no text>	night, lights, christmas, nightshot, nacht, nuit,notte, longexposure, noche, nocturna
	aheram, 0505 sarahc, moo	portrait, bw, blackandwhite, woman, people, faces, girl,blackwhite, person, man
	unseulpixel, naturey crap	fall, autumn, trees, leaves, foliage, forest, woods, branches, path

Input Text	2 nearest neighbours to generated image features	
nature, hill scenery, green clouds		
flower, nature, green, flowers, petal, petals, bud		
blue, red, art, artwork, painted, paint, artistic surreal, gallery bleu		
bw, blackandwhite, noiretblanc, biancoenero blancoynegro		

Recognition Results

- Multimodal Inputs (images + text), 38 classes.

Learning Algorithm	Mean Average Precision
Image-text SVM	0.475
Image-text LDA	0.492
Multimodal DBM	0.587

- Unimodal Inputs (images only).

Learning Algorithm	Mean Average Precision
Image-SVM	0.375
Image-LDA	0.315
Image DBN	0.452

Retrieval Results

Multimodal Query

hongkong, causewaybay, shoppingcentre, building, mall

me, myself, eyes, blue, hair

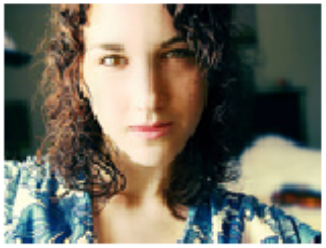
Top 4 retrieved results

howell, bridge, genesee, river, rochester, downtown, building

london, uk, night, skyline, river, thames, lights, bridge

edinburgh, scotland, dusk, bank

arcoiris, fincadehierro, lluvia, sannicolas, valencia



urban, me, abigfave, fiveflickrfavs

trisha, mynewcamera, lake, field, girl

me, ofme, self, selfportrait

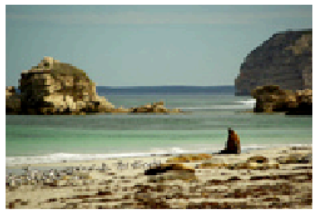
pink, prettyinpink, explored

Pattern Completion

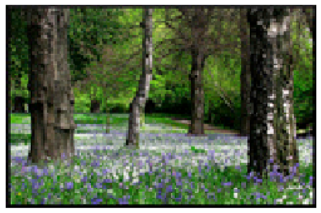
Given a test image, we generate associated text – achieve far better classification results.

landscape, scenery, hills,landscapes, scenic, land, canyon, roadtrip, place, tourism

portrait, black, white, girl, expression, lady, look, blonde, eyes, gorgeous



beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves

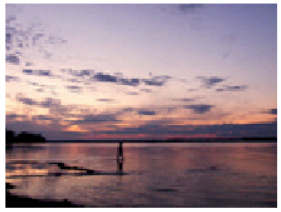


woods, breathtaking, hills, scenery, alone, mist, fields, bush, branches

sky, clouds landscape, hills, scenery, horizon, fields, landscapes, scenic, sun

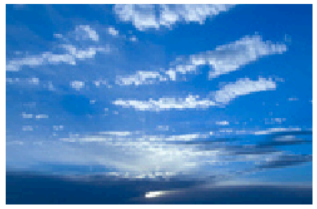
night, city urban, cityscape traffic, notte, skyline, lights, streets, skyscraper

car, engine, auto, supercar, ferrari, fast, gt, jason, parking, automobile



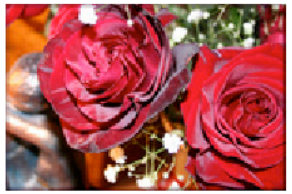
sunset, twilight, strand, wave, breathtaking, horizon, shore, seascape, surf, scenery

sky, blue, clouds, horizon, céu, twilight, azul, bleu, wave, sunset



sky, clouds, blue, horizon, céu, sunset, hills, twilight, bluesky, breathtaking

structure, facade, place, landmark, industry, skyscraper, tripod, royal, parking, 1910s



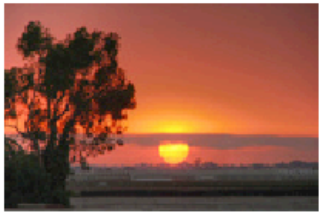
red, rouge, rosso, rot, catchycolors, gift, shiny, rojo, vivid, soft

Pattern Completion

blue, art, artwork, artistic surreal, expression, original, artist gallery, patterns

expression, lady, look, human, gorgeous, boys, portrait, person, blonde, sitting

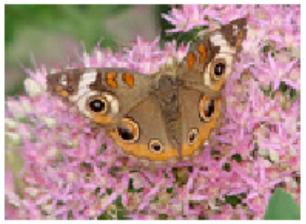
jason, video, supercar, fast, john, engine, gt, ferrari, collections, german



sunset, twilight, hills, breathtaking scenery, horizon, landscapes, calm vob, land

breathtaking christmastree, expression, newyear experiment photowalk, grain, 1910s

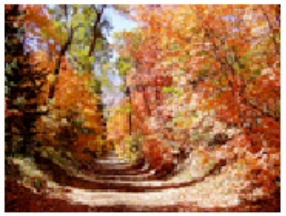
car, engine auto, fast ferrari supercar, jason, gt parking automobile



flower, pink, flowers, petals, macroflowerlovers, petal, flowerotica, floral, roses

winter, snow, frozen, frost, december, cold, hiver, ice, inverno, neige

place, german, english, john, royal, breathtaking, 1910s, collections, video, bill



fall, breathtaking, leaves, branches, autumn, woods, alone, branch, christmastree, frost

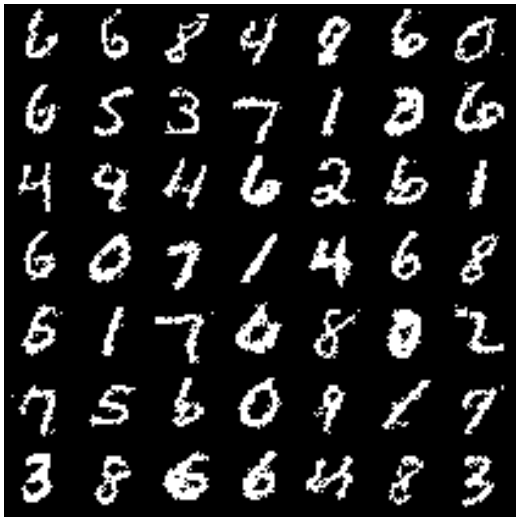
expression, video, artistic, john, collections, weird, human, magic, gorgeous, newyear

newyear, video, 1910s, christmastree, english, john, weird,gift, german, collections

Model Selection

How to choose the number of layers and the number of hidden units?

More generally, how can we choose between models?



DBM samples

Mixture of Bernoulli's

Goal: Compare $P(\mathbf{v})$ on the validation $P(\mathbf{v}) = P(\mathbf{v})^*/\mathcal{Z}$

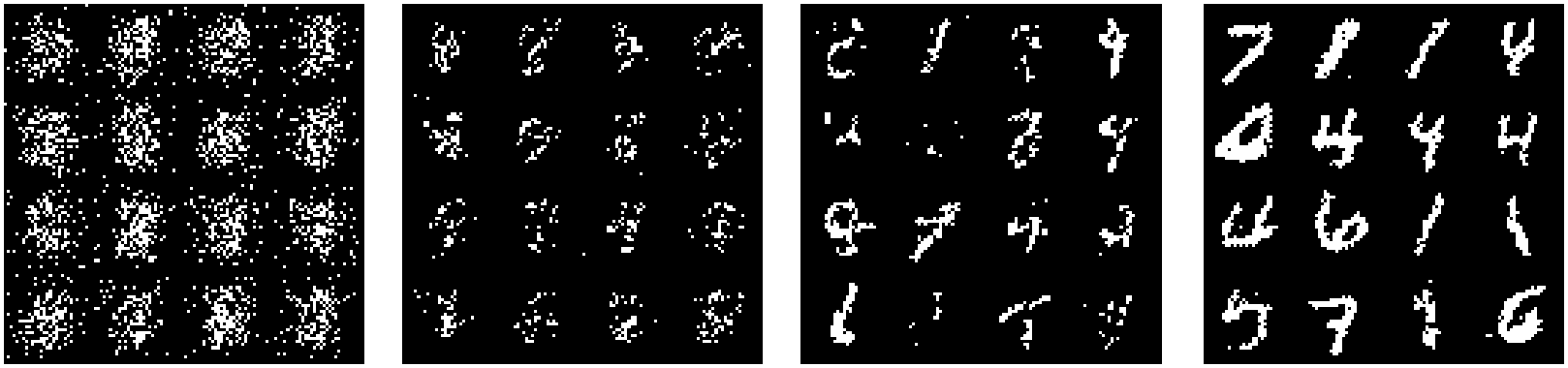
Need an estimate of partition function $\mathcal{Z}$

Model Selection

MCMC-based algorithm based on Annealed Importance Sampling to estimate partition function of a DBM model.

$$P_\theta(\mathbf{v};\beta) = \frac{1}{\mathcal{Z}(\beta)} P_\theta(\mathbf{v})^\beta \pi(\mathbf{v})^{(1-\beta)}$$

$\beta = 0$ $\beta = 0.5$ $\beta = 0.8$ $\beta = 1.0$

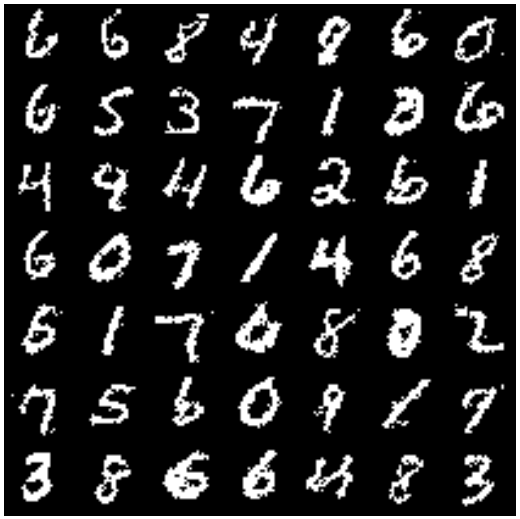


$$\frac{\mathcal{Z}(1)}{\mathcal{Z}(0)} = \frac{\mathcal{Z}(\beta_1)}{\mathcal{Z}(0)} \cdot \frac{\mathcal{Z}(\beta_2)}{\mathcal{Z}(\beta_1)} \cdot \frac{\mathcal{Z}(\beta_3)}{\mathcal{Z}(\beta_2)} \cdot \frac{\mathcal{Z}(\beta_4)}{\mathcal{Z}(\beta_3)} \cdot \frac{\mathcal{Z}(1)}{\mathcal{Z}(\beta_4)}$$

Annealing, or Tempering: $1/\beta =$ "temperature"

(Salakhutdinov & Murray, ICML 2008, Salakhutdinov 2008)

Model Selection



DBM samples

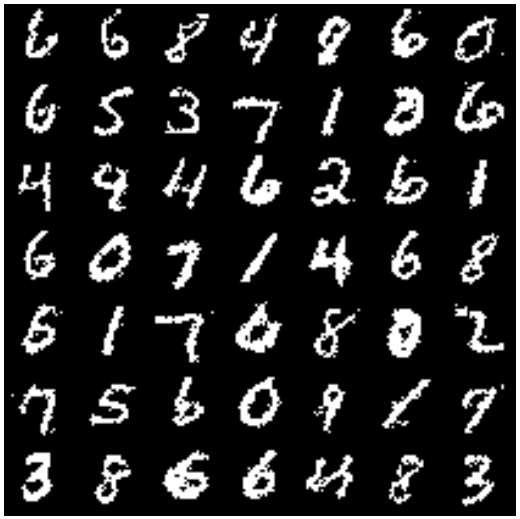
Mixture of Bernoulli's

MoB, test log-probability: -137.64 nats/digit

DBM, test log-probability: -85.97 nats/digit

Difference of over 50 nats is striking!

Model Selection



DBM samples

Mixture of Bernoulli's

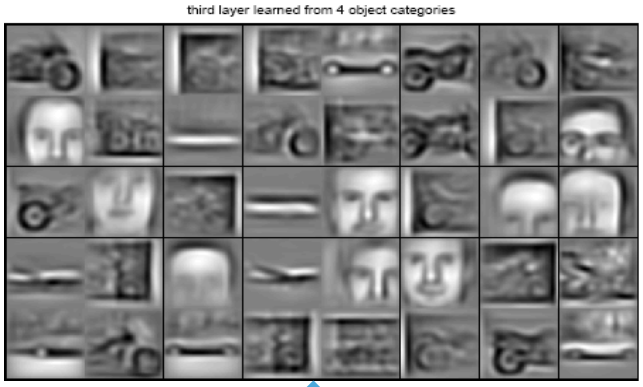
MoB, te

DBM, te

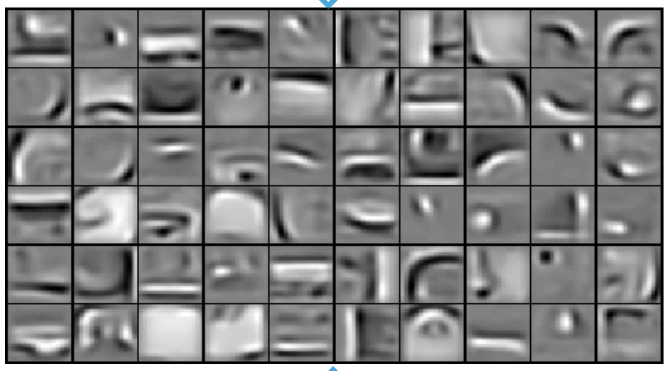
We will come back to advanced MCMC techniques later.

Difference of over 50 nats is striking!

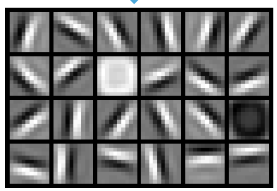
Learning Part-based Hierarchy



Object parts.



Combination of edges.



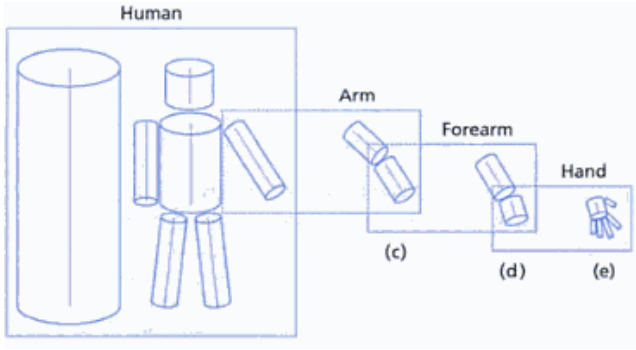
Trained from multiple classes
(cars, faces, motorbikes, airplanes).

Lee et.al., ICML 2009

Learning Hierarchical Representations

Deep Boltzmann Machines:

Learning Hierarchical Structure in Features: edges, combination of edges.



- Performs well in many application domains
- Combines bottom and top-down
- Fast Inference: fraction of a second
- Learning scales to millions of examples

Many examples, few categories

Next: Few examples, many categories – One-Shot Learning

One-shot Learning

How can we learn a novel concept – a high dimensional statistical object – from few examples.

(Lake, Salakhutdinov, Gross, Tenenbaum, CogSci 2011)

Traditional Supervised Learning

Segway

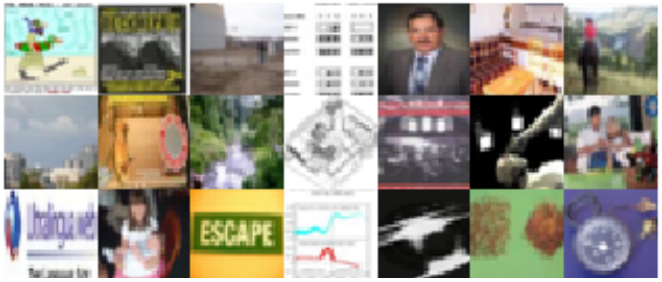
Motorcycle

Test:
What is this?

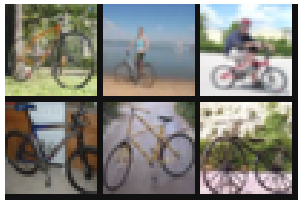
Learning to Transfer

Background Knowledge

Millions of unlabeled images



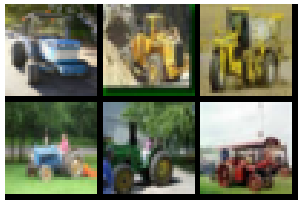
Some labeled images



Bicycle

Dolphin

Elephant



Tractor

Learn to Transfer Knowledge

Learn novel concept from one example

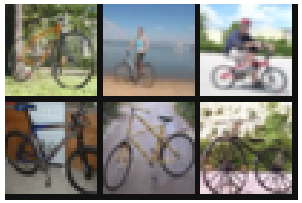
Test:
What is this?

Learning to Transfer

Background Knowledge

Millions of unlabeled images

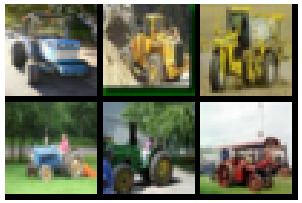
Some labeled images



Bicycle

Dolphin

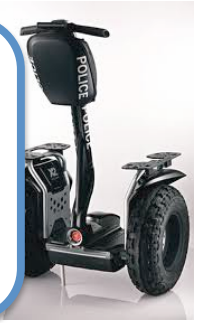
Elephant



Tractor

Key problem in computer vision, speech perception, natural language processing, and many other domains.

Learn to Transfer Knowledge



Learn novel concept from one example

Test:
What is this?

Thank you

Code for learning RBMs, DBNs, and DBMs is available at:
http://www.utstat.toronto.edu/~rsalakhu/code.html